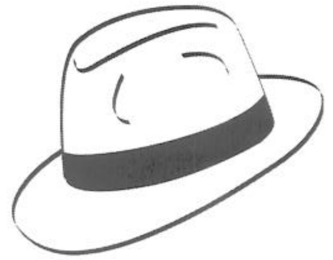
the HACKADEMY SCHOOL



Polycopié de cours

Cours Newbie+

Une publication The Hackademy School

Une publication **The Hackademy School**
© 2001-2002-2003 DMP

VISIT…

LANZAROTE Caliente.com

Bienvenue dans ce second volet des cours par correspondance de The Hackademy School

NOUS Y ÉTUDIERONS :

- Le Sniffing réseau — P. 4
- Tunneling — P. 10
- Analyse de paquets réseaux — P. 14
- L'altération de données réseaux — P. 17
- L'IP Spoofing — P. 20
- L'idle host scanning — P. 30
- Les Network Mappers — P. 34
- Les DNS Queries — P. 39
- Les attaques DoS — P. 47
- Le Fingerprinting — P. 51
- Introduction aux vulnérabilités PHP — P. 59
- Vulnérabilités CGI — P. 68

Ces chapitres constituent l'ensemble du cours Newbie +. Les premiers chapitres sont les plus "costauds", il vous sera donc peut-être nécessaire de vous replonger sur vos premiers cours - surtout sur les parties consacrées aux protocoles - mais ne vous inquiétez pas. De nombreuses captures d'écran et instructions "pas-à-pas" viendront vous soutenir dans la lourde quête que vous vous êtes fixée, à savoir l'apprentissage.

Petite indication nécessaire à la lecture de ce cours : de nombreux termes techniques sont à la fois en Anglais et en Français... Ce n'est pas parce que l'auteur perd la tête au fur et à mesure qu'il progresse dans son cours, c'est simplement une méthode de sensibilisation et d'initiation aux termes techniques courants sous leurs formes françaises et américaines. Ne vous inquiétez pas, tout est rédigé de sorte que vous ne perdiez pas le fil des explications...

Une publication The Hackademy School
© 2001-2002-2003 DMP

<u>IMPORTANT:</u>

Ce polycopié de cours de The Hackademy School a pour objectif unique de contribuer à une meilleure compréhension des risques de sécurité liés à l'usage de l'outil informatique, et par là de permettre de s'en protéger plus efficacement. Il sera utile aux administrateurs système et réseau, aux développeurs, et plus généralement à toute personne utilisant Internet chez elle ou au travail. Si vous êtes soucieux de comprendre comment un pirate pourrait tenter de vous attaquer afin d'être à même de déjouer ses tentatives, ce cours est fait pour vous. Cependant, aucune garantie n'est donnée que ce contenu va vous permettre de vous protéger efficacement ou de manière "ultime", car ce n'est pas le cas et ça ne le sera jamais. De plus, le contenu de ce cours de niveau "débutant éclairé" est loin de couvrir le sujet de manière exhaustive: nous vous détaillons des méthodes d'attaque courantes, et nous vous fournissons des pistes pour vous en protéger. Nous vous donnons les bases nécessaires, à vous d'approfondir les points qui vous concernent directement, soit grâce à The Hackademy School, soit en faisant des recherches sur Internet.
Ce cours peut présenter des erreurs ou omissions susceptibles de vous porter prédudice. The Hackademy et DMP ne sauraient être tenus pour responsables des dommages éventuels causés par une application des méthodes présentées ici sur un système. Merci de nous prévenir si vous constatez de telles erreurs.
Il est formellement interdit par la loi d'appliquer les techniques d'attaque présentées dans ce cours sur un système que vous ne possédez pas. Vous pouvez cependant les appliquer sur vos systèmes informatiques à des fins de tests de vulnérabilité, en gardant à l'esprit que cela présente toujours des risques que vous assumerez seul.

Loi N° 88-19 du 5 Janvier 1988 relative à la fraude informatique. Extraits donnés pour illustration.

Accès ou maintien frauduleux dans un système informatique :
2 mois à 1 an de prison,
2 000 à 50 000 francs d'amende.

Accès ou maintien frauduleux dans un système informatique avec dommages involontaires : modification ou suppression de données, altération du fonctionnement du système
2 mois à 2 ans de prison,
10 000 à 100 000 francs d'amende.

Entrave volontaire au fonctionnement d'un système informatique :
3 mois à 3 ans de prison,
10 000 à 100 000 francs d'amende.

Introduction, suppression, modification intentionnelles de données :
3 mois à 3 ans de prison,
2 000 à 500 000 francs d'amende.

Suppression, modification intentionnelles du mode de traitement, des transmissions de données :
3 mois à 3 ans de prison,
2 000 à 500 000 francs d'amende.

Falsification de document informatique, usage de document falsifié :
1 an à 5 ans de prison,
20 000 à 2 000 000 francs d'amende.

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

LES COURS PAR CORRESPONDANCE DE THE HACKADEMY SCHOOL

I - Le sniffing réseau

1 - Approche théorique

Le sniffing est une technique qui peut se résumer en une phrase, à comprendre puis à apprendre par cœur : *" Technique d'espionnage consistant à copier les informations contenues dans des paquets réseaux, sans en altérer l'acheminement ou leur formation. "* Vous n'avez rien compris ? Eclaircissons.

Lorsqu'une machine A va contacter une machine C, elle va faire transiter les données par une machine intermédiaire, une machine B.

A -----> B -----> C

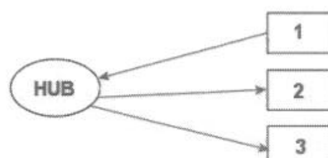
Pour atteindre C, A doit faire transiter le paquet par B. B pourrait être sous contrôle d'un pirate. Celui-ci pourrait alors malicieusement capter les données qui transitent entre A et C et en faire une copie pour une analyse ultérieure. Pour le pirate, cette méthode est celle du "sniffing".

Le sniffing permet la capture des données qui forment les paquets réseaux, à savoir les différentes options et variables incrémentées dans les paquets (adresse IP Source, adresse IP Destination, Flags, etc.), aussi bien que les données usuelles (codes sources de pages web, logins et mots de passes, commandes envoyées à un serveur, etc.). Le sniffing se base uniquement sur des principes logiciels, et non pas matériels. Ce qui veut dire qu'un simple logiciel suffit à mettre en place une technique de sniffing. La raison d'être du sniffing vient d'une faiblesse de sécurité sur la grande majorité des protocoles. Les données transmises grâce à la plupart des protocoles courants (TCP, HTTP, FTP, SMTP, ...) ne sont pas cryptées. Ainsi leur intégrité n'est pas garantie puisque n'importe qui, situé à un nœud de communication réseau, peut, avec un peu d'habileté, copier l'ensemble des données réseaux qui circulent par sa machine. Dans certains environnements réseaux, il n'est même pas nécessaire de faire partie d'un point du système de relai (routeur, passerelle, etc.) pour entreprendre du sniffing de réseau...

2 - Environnements réseaux : facilités et limites du sniffing.

Le sniffing est une méthode qui se limite déjà aux zones que le pirate contrôle. Il ne faut pas imaginer que l'on peut analyser le trafic réseau d'une machine distante. On ne peut sniffer que sur des machines qui offrent suffisamment d'accès au pirate pour installer un "sniffer" (le "sniffer" est le logiciel qui permet de faire du "sniffing"). Le sniffer va tourner comme une application ou un service sur un système, capturant les données qui transitent par la machine sur laquelle il s'est installé. Un sniffer n'est donc pas un virus ! De plus, une très grande partie des sniffers ne sont pas assimilables à des chevaux de Troie, ils sont donc légaux et non détectés par les anti-virus. Pour couronner le tout, la majorité d'entre eux sont gratuits. Vous trouverez donc des sniffers facilement et n'aurez aucun problème lié à l'utilisation ou aux risques que cela pourrait impliquer. Un sniffer n'est pas dangereux pour un système, mais juste pour la confidentialité des données réseaux que gère le système, si, et seulement si, on en fait une utilisation malveillante. Les sniffers sont aussi appelés "analyseurs réseaux" et sont utilisés régulièrement par les administrateurs pour résoudre des problèmes sur le réseau. Ils sont également utiles pour surveiller d'éventuelles intrusions de pirates !

Dans certains environnements réseaux, il n'est pas nécessaire d'avoir le contrôle d'une machine relai pour espionner l'ensemble des flux de données. C'est par exemple le cas des architectures réseaux se structurant autour d'un HUB. Tout l'ensemble du réseau formé autour du hub est vulnérable à une technique de type Sniffing, en raison du mode de fonctionnement technique du Hub. Un Hub a en effet la caractéristique de renvoyer à l'ensemble des machines qu'il connecte tous les paquets qu'il reçoit. Ainsi il n'a pas à se soucier de savoir si, oui ou non, tel paquet était destiné à telle machine sur le réseau : il laisse aux machines réceptrices le soin d'effacer les paquets qui ne leurs sont pas destinés.

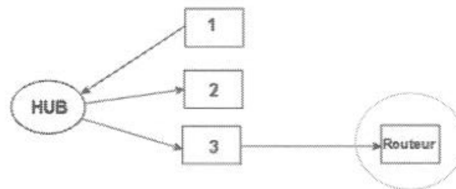


Ici la machine 1 envoye un paquet sur le réseau (peu nous importe sa destination.
Le HUB, pour le relayer, le renvoie à toutes les autres machines du réseau.

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

Sur notre schéma, la machine 2 pourrait être celle d'un pirate. Celui-ci, équipé d'un sniffer, pourrait alors espionner tout le trafic réseau de toutes les machines reliées au HUB, car tous les paquets lui sont renvoyés.

Remarque : Il n'est pas possible de sniffer une branche d'un réseau qui ne relaye pas directement les machines à espionner.



Ici, la connexion entre le routeur et la machine 3 n'est pas sniffable par le reste du réseau.

Sur cet exemple-ci, les machines 1 et 2 ne peuvent sniffer la connexion établie entre la machine 3 et le routeur, car elles n'ont pas accès à cette branche du réseau, ou n'ont pas de contrôle sur le routeur ou la machine 3. A partir de ce shéma, on comprend mieux pourquoi, lorsque l'on est connecté à Internet, on ne peut sniffer que le trafic réseau qui passe par sa machine, et pas celui passant par les machines d'autres internautes ou entreprises. Cette éventualité n'arrive qu'en réseau local, dans certaines architectures réseaux particulières, comme celle que l'on vient de voir concernant le HUB.

A noter aussi que la même architecure réseau pourrait précisément se former non pas autour d'un Hub, mais autour d'un Switch. Ces deux engins ont exactement la même fonctionnalité : relier entre elles différentes machines pour former un réseau. Toutefois, certaines caractéristiques les différencient :

- Le prix : le prix d'un Hub (plusieurs centaines de francs) est bien moins élevé que celui d'un Switch. Les prix varient de 80 € pour les plus modestes d'entre eux que l'on réservera aux particuliers à plusieurs centaines d'euros pour ceux munis d'un grand nombre de ports et qui eux seront idéaux pour un usage en entreprise.
- Le mode de fonctionnement : le Switch sait où se situe telle ou telle machine sur le réseau, il n'agit donc pas comme un HUB et renvoie à la machine demandée uniquement les paquets qui lui sont destinés.

Cela implique un petit nombre de choses qui le différencient du Hub lors de son utilisation :

- Les surcharges réseaux sont beaucoup moins fréquentes, et le trafic réseau lui-même est atténué ;
- Le Switch apparait comme une solution de sécurité par rapport au Hub... en héorie du moins. Car en pratique, méfiez-vous: les techniques d'ARP spoofing permettent de contourner cette sécurité et de sniffer sur un réseau switché. L'emploi de protocoles sûrs (chiffrés par exemple avec une SSL) reste donc nécessaire.

3 - Approche pratique : installation d'un sniffer.

Installer un sniffer n'a rien de sorcier. L'intérêt est de pouvoir analyser le traffic réseau entrant et sortant, pour mieux comprendre le fonctionnement des protocoles, et être capables de détecter si un individu mal intentionné fait un usage illicite de votre réseau ! Ce qu'il vous faut avant tout, c'est trouver un sniffer valable, pas cher (nous en prendrons un gratuit), et simple d'utilisation. Vous rendre sur http://www.ethereal.com vous semble soudain irresistible. Dans la section download, vous trouverez Etheral en libre téléchargement pour différents systèmes d'exploitation. Windows se situe en bas du tableau. Téléchargez la dernière version d'Ethereal, mais avant de commencer, il vous faudra installer WinPCAP. qui est un jeu de librairies, pour tout système Windows, qui permet de faire du sniffing. Vous le trouverez sur http://winpcap.polito.it.

Installation de WinPCAP :

Sur cet exemple, c'est "l'auto-installer" de WinPCAP version 2.3 qui a été téléchargé. Rien de bien complexe :

1. Lancement de l'application ;
2. Acceptation de la license (après l'avoir lu... théoriquement...) ;)
3. Des clics répétés sur "Next" jusqu'à ce qu'on arrive au bouton "Ok".

Par précaution, redémarrez votre machine.

© 2001-2002-2003 **DMP**

Installation d'Ethereal :

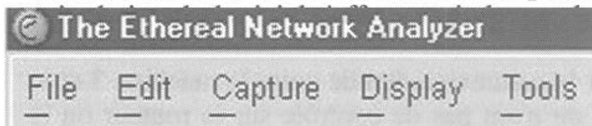
Une fois votre machine redémarrée, lancez l'installation d'Ethereal. Sur cet exemple, c'est Ethereal version 0.9.3 qui a été téléchargé.

1. Accepter la licence après une longue et (in?)utile lecture ;
2. Choisir les composants à installer. Dans ce cas, vous pouvez tous les laisser, ce ne sera pas un problème ;
3. Poursuivre les étapes d'installation jusqu'à la fin.

Ca y est, vous avez installé Ethereal, et les composants nécessaires pour pratiquer, dans les dix secondes à venir, le sniffing.

4 - Approche pratique : utilisation d'un sniffer.

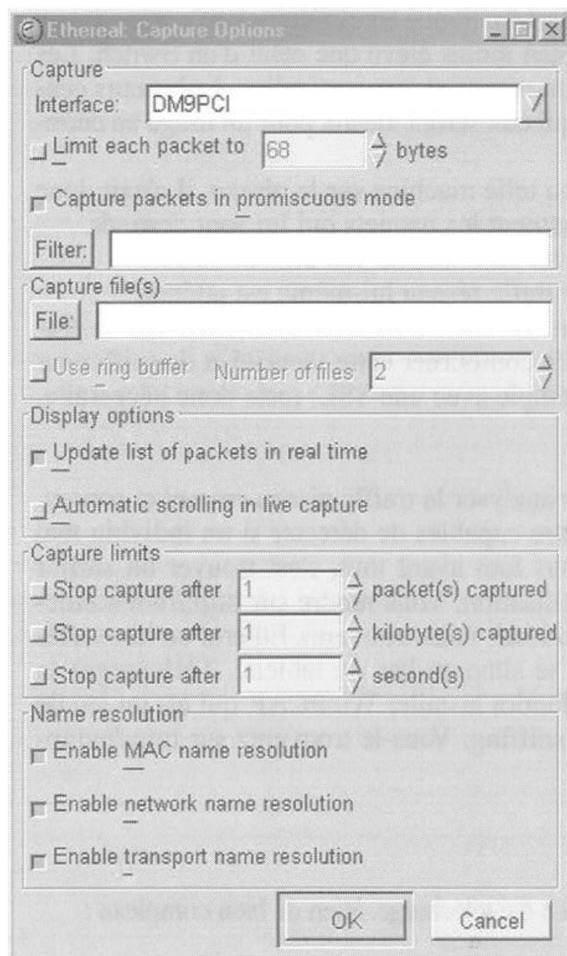
Lancez Ethereal. Nous allons travailler, pour cette partie du cours, sur Ethereal 0.9.3, sous Windows. La



ts en haut de fenêtre.

Nous allons commencer par faire des sessions de sniffing brutes avant de s'intéresser aux puissantes options de capture.

1. Cliquez sur Capture :
2. Cliquez sur Start.





Une fenêtre de paramétrage de la session de capture s'ouvre. Parmi les différentes fonctionnalités à portée de main, notez que vous pouvez spécifier dans "Interface" le périphérique à sniffer. Si vous avez plusieurs cartes réseaux, vous pouvez sniffer le traffic sur la carte de votre choix. Parfois il arrive que plusieurs options se présentent dans le choix de l'interface. Certaines d'entre elles ne marchent pas, mais si votre réseau fonctionne, au moins une interface marche à coup sûr. Dans cette exemple, DM9PCI correspond à la carte réseau de l'auteur. Il n'y a pas lieu de s'inquiéter si des données différentes apparaissent dans votre cas.

Parmi les autres options qu'il est intéressant de modifier, notons les "Display options" et "Capture limits". En particulier, elles vous permettent de suivre en temps réel le sniffing des paquets "Update list of packets in real time" (option recommandée), d'autres de faire glisser l'ascenseur automatiquement au fur et à mesu-

© 2001-2002-2003 **DMP**

re que s'affichent les paquets "Automatic scrolling in live capture", ou encore d'imposer des limites à la session de sniffing en la faisant s'interrompre selon des critères bien définis (après un certain nombre de paquets capturés, une certaine taille de données capturées, ou un certain laps de temps) dans "Capture limits".

Pour votre premier essai, activez les options :

- Capture packet in promiscuous mode ;
- Update list of packets in real time ;
- Les 3 options de "Name resolution".

NB : Par défaut, une carte ne récupérera que les paquets lui étant adressés, les autres étant détruits. Pour lire les paquets à destination des autres ordinateurs, le pirate devra mettre la carte dans un mode spécial, le "mode promiscuous". La carte capturera désormais tous les paquets. Il est toutefois important de noter que puisque celle-ci doit être manipulée à un très bas niveau pour ce changement de mode de fonctionnement, les privilèges doivent être suffisants. Tout dépendra de la configuration du système mais il est vraisemblable qu'un utilisateur simple n'aura pas ces droits. Il ne pourra, de ce fait, que sniffer les données à destination de sa propre machine (ou émises par celle-ci).

Par la suite, vous pourrez tester les diverses fonctionnalités et mieux appréhender le fonctionnement du logiciel en testant, par vous-même, les autres options de lancement d'une capture.

Ethereal: Cap...

Total	145	(100,0%)
SCTP	0	(0,0%)
TCP	120	(82,8%)
UDP	16	(11,0%)
ICMP	9	(6,2%)
OSPF	0	(0,0%)
GRE	0	(0,0%)
NetBIOS	0	(0,0%)
IPX	0	(0,0%)
VINES	0	(0,0%)
Other	0	(0,0%)

Stop

Validez par "Ok". La capture commence.

La fenêtre du statut de la capture indique combien de paquets ont déja été sniffés, ainsi que la plupart des protocoles courants auxquels correspondent ces paquets. C'est aussi cette fenêtre qui sert à mettre fin à la session de capture.

Vous allez voir défiler de nombreuses informations dans la fenêtre principale d'Ethereal. A première vue, tout cela semble très chaotique. Voyons comment soutirer des informations pertinentes et utiles de cette masse de données.

Utilisation des onglets :

Vous pouvez classer les données reçues par ordre d'apparition, c'est-à-dire dans l'ordre où le sniffer les a reçues. Vous pouvez aussi les classer par adresse source (l'adresse de l'émetteur) ou par adresse de destination (l'adresse de la cible où sont destinés les paquets réseaux). Enfin, vous pouvez les classer par protocoles, et par les informations qu'ils contiennent.

- **Onglets correspondants au classement par**
  - o Ordre d'apparition : No.
  - o Adresse IP Source : Source
  - o Adresse IP de destination : Destination
  - o Protocoles : Protocol

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

o Informations contenues dans les paquets Info

o L'onglet "Time" ne vous servira probablement pas beaucoup, que ce soit dans l'immédiat, ou dans des sessions de capture à venir.

Ainsi, une fois votre session de capture lancée ou achevée, vous pouvez rechercher tous les paquets que votre machine a émis durant la capture. Il vous suffira pour cela de faire :

1. Un clic sur l'onglet "Source" ;
2. De rechercher l'adresse de votre machine dans la liste des paquets ainsi triés.

Pour visualiser les données que votre machine a reçues, faites :

1. Un clic sur l'onglet "Destination" ;
2. Recherchez l'adresse IP de votre machine dans la colonne correspondant à "Destination".

En quelques clics, vous pouvez même pousser le classement de façon plus approfondie. Par exemple, si vous voulez tous les paquets HTTP que votre machine a émis :

1. Cliquez sur l'onglet "Source"
2. Cliquez sur l'onglet "Protocol"
3. Recherchez tous les paquets HTTP (colonne "Protocol") émis par votre machine, associés à l'adresse IP source (colonne "Source") de votre machine.

Vous pourrez varier les modes de classement à votre gré, vous vous apercevrez bien vite que manipuler le logiciel par onglets est intuitif, simple, mais parfois trop.

Utilisation de l'option "Follow TCP Stream" :

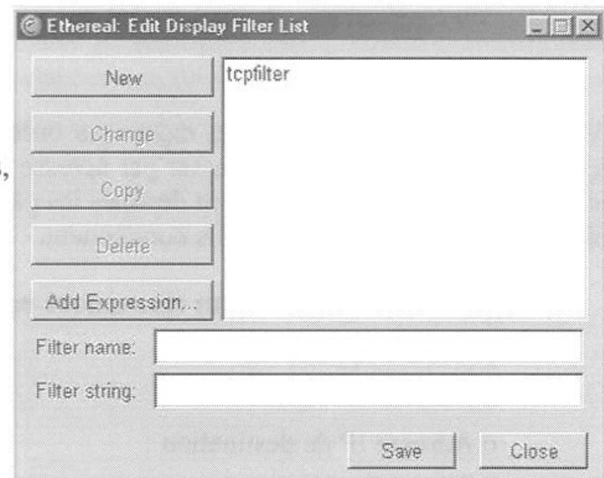
L'option "Follow TCP Stream" est bien pratique. Elle permet d'isoler une "conversation", un échange de données, entre deux machines bien précises. Ainsi, si dans les résultats de votre session de capture, vous souhaitez zoomer rapidement sur une communication, faites comme suit :

1. Sélectionnez le paquet émis d'une machine à une autre. Par exemple, cliquez sur une ligne correspondant à un paquet émis de la machine A à B ;
2. Faites un clic droit ;
3. Cliquez sur "Follow TCP Stream"
4. S'ouvre alors une nouvelle fenêtre qui ne contient que le contenu de l'échange des données de traitement qu'il y a eu entre les deux machines. Un jeu de couleurs est mis en place pour que l'on distingue les données appartenant à telle ou telle machine.
5. Dans le bas de la fenêtre principale du logiciel, dans la zone "Filter", s'affichent les informations de filtrage qui permettent un tri efficace de l'affichage de l'écran pour mettre en valeur l'échange qui nous intéresse. Il faut bien maîtriser le logiciel pour comprendre parfaitement la signification de ce filtrage. Ici, ce n'est pas nécessaire.

Utilisation des filtres :

Comme vu précédemment, les filtres vont vous permettre de faire de la discrimination parmi les informations capturées ou en cours de capture. Vous pouvez créer des options de filtrage via l'interface désignée du logiciel :

- Dans le logiciel, cliquez sur l'onglet "Edit" ;
- Cliquez ensuite sur "Display Filters" ;
- S'ouvre alors une fenêtre qui va vous permettre de créer des filtres.
- Nous allons voir, par quelques exemples précis, comment gérer efficacement vos filtres.

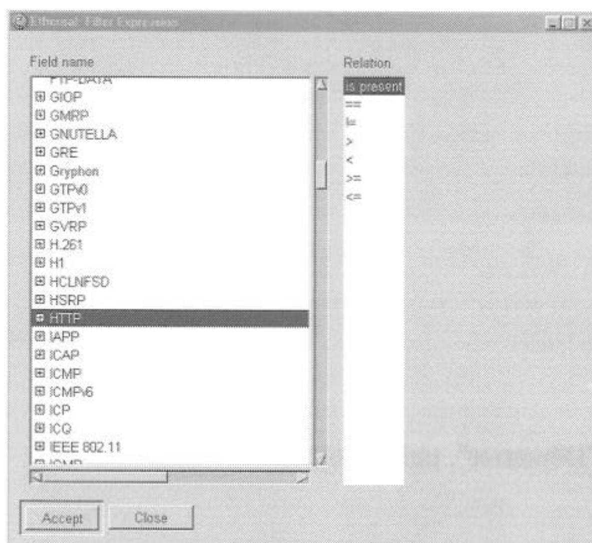


Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

Exemple 1 : filtrer uniquement des paquets HTTP.

Vous voilà donc face à votre fenêtre de gestion des filtres, et pour pouvoir maîtriser cet environnement, qui, vous allez le voir, est en fait très simple d'utilisation, nous allons prendre un exemple concret. Supposons que vous ne souhaitiez filtrer que les données HTTP, afin que lors du lancement d'une session de capture, n'apparaissent et ne soient consultables que celles-ci.

1. Dans la fenêtre "Ethereal : Edit Display Filter List", cliquez sur "Add Expression" ;
2. S'ouvre alors une fenêtre qui liste tous les protocoles reconnus par le logiciel ;
3. Allez sur HTTP ;



4. Cliquez sur HTTP, il est inutile après d'aller dans la colonne "Relation". Nous verrons dans un autre exemple à quoi celle-ci peut nous servir ;
5. Cliquez sur "Accept" ;
6. Vous revenez à la fenêtre de gestion de filtres. Dans "Filter String" a dû apparaître "http" ;
7. Donnez un nom ("Filter Name") à ce filtre. Vous pouvez lui donner "http" si vous voulez que ce soit évocateur ;
8. Cliquez sur "New" ;
9. Dans la liste de vos filtres, est venu s'en greffer un nouveau, celui que vous venez de créer ;
10. Cliquez sur "Save" puis sur "Close".

Vous venez de créer un nouveau filtre. Dans vos futures sessions de capture, vous allez pouvoir l'appliquer de la façon suivante :

1. Lancez une nouvelle session de capture ;
2. Cliquez sur le bouton Filter en bas du logiciel ;



3. Sélectionnez le filtre approprié ;
4. Cliquez sur "Apply" ;

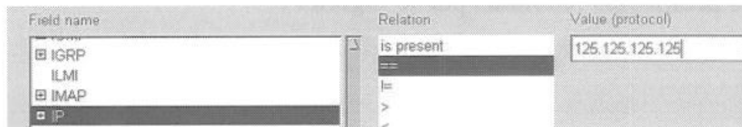
Mais attention, notez bien que les filtres ne servent qu'à discriminer les informations qui s'affichent pour plus de lisibilité. Dans une session de capture, Ethereal va tout de même capturer tous les paquets. Loin d'être un défaut, cela peut vous permettre de retourner à un affichage complet des informations. Etudions maintenant un deuxième exemple de création de filtres, un peu plus approfondi.

© 2001-2002-2003 **DMP**

Exemple 2 : filtrer par adresses IP.

Ce que l'on aimerait faire cette fois, c'est de ne visualiser que les paquets réseaux spécifiques à une adresse IP. On va supposer que vous vouliez ne filtrer que les paquets réseaux émis en direction de l'adresse IP 125.125.125.125.

1. Dans la fenêtre "Ethereal : Edit Display Filter List", cliquez sur "Add Expression" ;
2. S'ouvre alors la même fenêtre qui liste tous les protocoles reconnus par le logiciel ;
3. Cliquez sur "IP" dans la liste, puis dans la colonne "Relation" cliquez sur le double signe égal ;
4. Rentrez alors l'adresse IP 125.125.125.125 dans la case adéquate "Value (protocol)" ;



5. Cliquez sur "Accept" ;
6. Donnez un nom au filtre, et cliquez sur "New" ;
7. Puis cliquez sur "Save" et "Close" ;
8. Lancez une nouvelle session de capture ;
9. Cliquez sur le bouton Filter en bas du logiciel ;



10. Sélectionnez le filtre approprié ;
11. Cliquez sur "Apply" ;
12. Ouvrez telnet.exe de la façon suivante : allez dans "Démarrer", puis "Exécuter" et tapez telnet 125.125.125.125 ;
13. Validez avec "OK" ;
14. Sur la fenêtre principale d'Ethereal s'affiche alors, normalement si tout s'est bien passé, les informations relatives à votre tentative de connexion vers 125.125.125.125, et uniquement celles-ci.

Avec un peu de pratique, vous pourrez mettre en place des configurations appropriées répondant à vos besoins. Mais ne mettre aucun filtre n'est pas un problème, pourvu que l'on arrive à lire avec habitude les informations qui s'affichent. La meilleure chose que vous puissiez faire pour vous entraîner est de sniffer des ouvertures de connexions à distance afin d'analyser les processus au niveau des protocoles. Le sniffing est le meilleur moyen de visualiser de façon pratique des informations théoriques sur les protocoles. Après ce qu'un pirate peut en faire... Vous vous l'imaginez facilement, uniquement par le peu de pratique effectuée.

II - Le tunneling

Le tunneling correspond à une encapsulation, lorsque l'on parle de protocoles. Ainsi, le protocole TCP est encapsulé dans le protocole IP. L'usage du tunneling consiste en une communication via un protocole donné, lequel, pour pouvoir être transmis sur le réseau, est inséré au sein d'autres protocoles.

Dans cette partie, nous allons vous montrer pourquoi il est très dangereux de supposer que les utilisateurs d'un réseau privé ne pourront pas avoir un accès complet à internet, même si vous mettez en place des règles drastiques de firewall. Un utilisateur malintentionné pourrait en effet tunneler n'importe quel protocole à travers le proxy web (protocole HTTP) pour contourner les règles de filtrage mises en place dans une entreprise. De même, un cheval de troie installé au sein de votre réseau pourrait communiquer avec le pirate sur internet grâce au tunneling ! Imaginons que les règles ne permettent que de surfer sur le Web, alors que l'utilisateur - ou le pirate - désire utiliser d'autres protocoles tels que FTP, IRC, POP3, SMTP, etc. Eh bien, c'est possible ! Dans certains cas, disposer d'une machine à l'extérieur du réseau protégé est nécesaire pour relayer la connection.

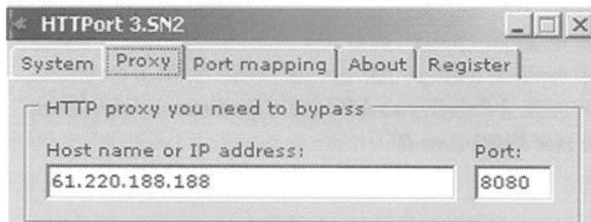
Pour comprendre le tunneling, nous allons avoir besoin de Httport et Htthost qui sont téléchargeables sur http://www.htthost.com. Httport permet d'outrepasser un proxy HTTP et de surfer en utilisant 2 proxies chainés l'un derrière l'autre. En chaînant 2 proxies, vos paquets en passant par le premier proxy vont changer d'IP source (votre IP va être remplacée par celle du proxy1), idem en passant le second (IPproxy1 changée par IPproxy2) avant d'arriver au serveur web distant (l'IP a donc été changée 2 fois). Cette méthode

© 2001-2002-2003 **DMP**

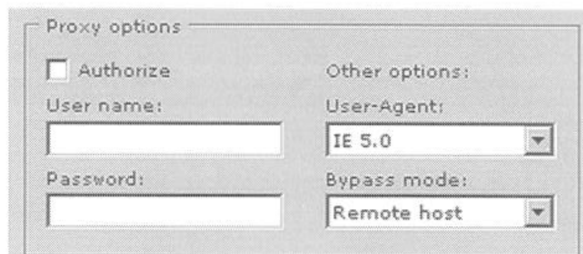
n'est pas réservée aux pirates: elle peut être utilisée de manière légitime pour protéger votre anonymat sur Internet.

Chaîner 2 proxies :
Nous choisissons deux proxies :
61.222.104.194:8080 --> 1
202.188.17.35:80 --> 2
Attention, ces adresses IP d'exemple ne sont pas des proxies valides. Vous trouverez une liste de proxies anonymes sur : http://www.multiproxy.org. Prenez soin de demander l'autorisation du propriétaire du proxy avant de l'utiliser, sans quoi votre utilisation pourrait être assimilé à une intrusion !

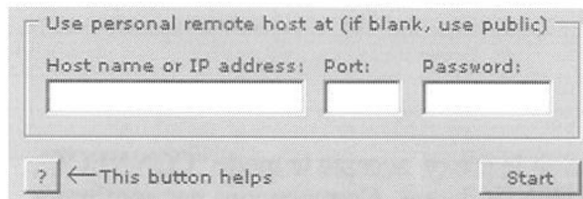
Lancez Httport, et clickez sur l'onglet "proxy".



Vous devez mettre dans cette partie l'IP et le port du 1er proxy à utiliser. Pour simuler le comportement d'un cheval de troie cherchant à accéder à Internet depuis votre réseau interne protégé (votre entreprise par exemple), mettez ici l'adresse IP du proxy web du réseau.

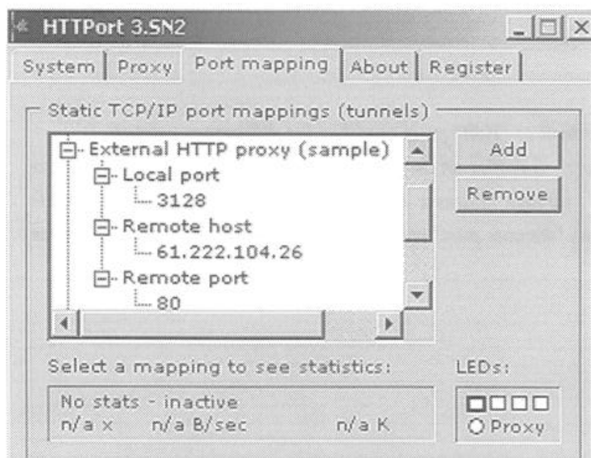


Cochez "Authorize" et remplissez les divers champs si vous utilisez un proxy demandant une identification.



Ici, laissez les champs vide (on utilisera cette option pour le tunneling dans la suite du cours).

Cliquez sur l'onglet "Port Mapping":
L'onglet port mapping permet d'ouvrir des ports sur notre ordinateur qui redirigeront directement les paquets vers le serveur spécifié dans "port mapping" au travers du proxy de l'entreprise défini dans l'onglet "proxy".



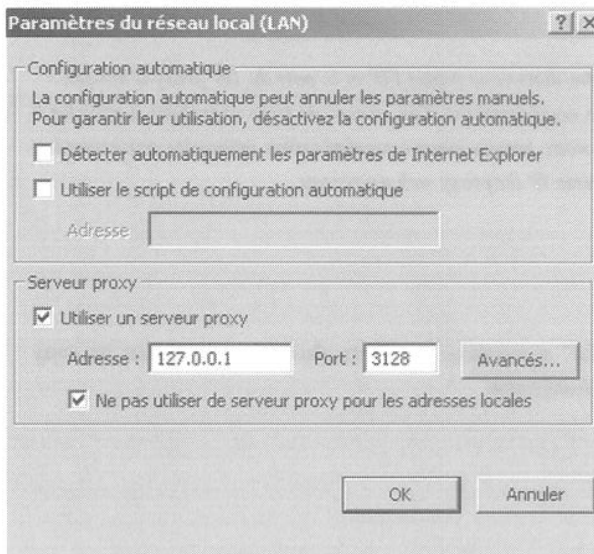
Vérifiez que vous avez bien "External HTTP proxy (sample)" dans "Static TCP/IP port mapping". C'est ici que l'on spécifiera le 2ème proxy à travers lequel on désire passer. Dans la capture ci-contre, on a défini le port 3128 ouvert sur notre machine pour qu'il se connecte sur le proxy 61.222.104.26 sur le port 80.

© 2001-2002-2003 **DMP**



Cochez la case "Run Socks server" (port 1080).

Retournez sur l'onglet "proxy" et cliquez sur "Start" pour activer Httport.
Il ne vous manque plus qu'à configurer votre Navigateur pour qu'il se connecte sur vous-même (127.0.0.1) sur le port 3128. Pour cela (sous IE) cliquez sur Outils --> Options Internet... --> Paramètres LAN.

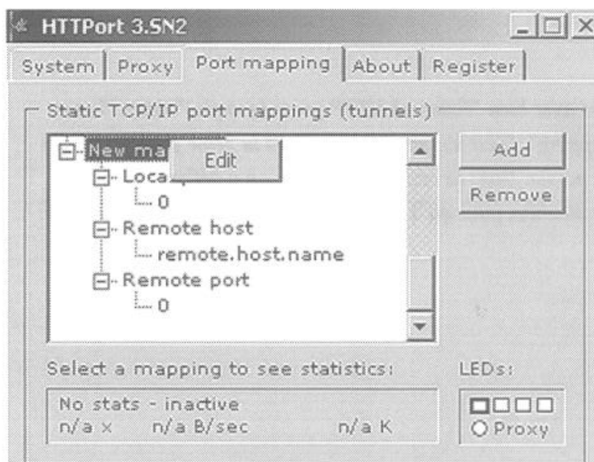


Cochez "Utiliser un serveur proxy", dans adresse mettez "127.0.0.1" et port "3128". De cette manière, à chaque fois que vous utiliserez IE, il se connectera à votre port 3128, à Proxy1, puis à Proxy2 et finalement au site distant dont vous aurez tapé l'URL dans IE.

Vous pouvez vérifier votre IP sur www.dmpfrance.com

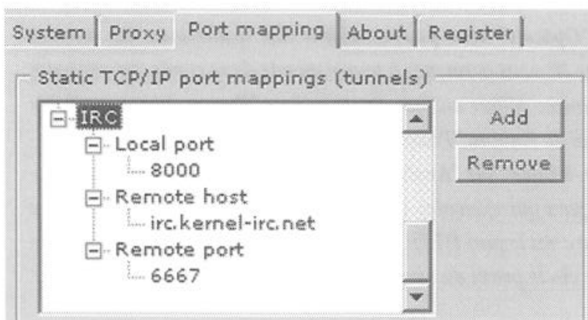
Le tunneling avec Httport et Htthost :

Pour utiliser le tunneling avec Httport, il y a deux solutions. Soit le proxy accepte le mode "CONNECT", soit il va falloir une machine relai à l'extérieur du réseau avec Htthost dessus. Commencons par configurer notre "port mapping" pour nous connecter via le proxy sur un serveur IRC par exemple. Vous pouvez télécharger un client IRC comme mIRC que vous pouvez trouver sur http://www.mirc.com. La connexion sur un channel IRC est une méthode privilégiée utilisée par plusieurs virus et chevaux de Troie pour se faire contrôlés à distance par le pirate, il est donc utile d'étudier la faisabilité de ce type de tunneling sur votre réseau.
Allez dans l'onglet "Port mapping" de Httport et cliquez sur add pour ajouter un nouveau mapping.



Faites un click droit et "Edit" pour modifier les différentes valeurs...
Dans "Local port", spécifiez un de vos ports libres, 8000 par exemple. Dans "Remote host" indiquez l'adresse du serveur IRC, par exemple irc.kernel-irc.net, puis dans "Remote port" mettez "6667" qui est le port par défaut des serveurs IRC.

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**



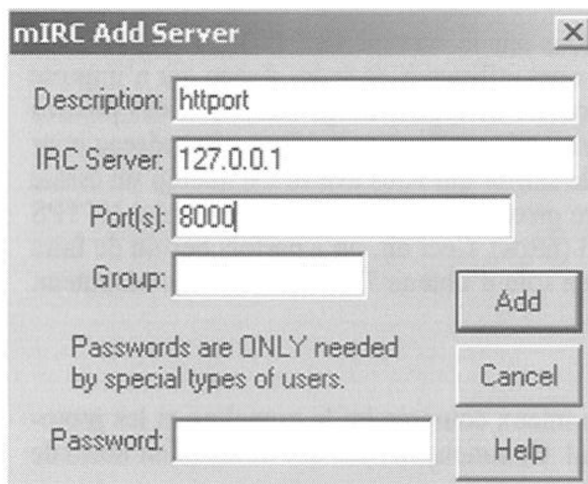
Une fois configuré, vous devriez avoir ceci.

Configurons à présent notre client IRC pour qu'il se connecte sur notre port local 8000, pour être redirigé ensuite directement par Httport au travers du proxy défini dans l'onglet "proxy".

Lancez mIRC... Cliquez sur "File" --> "Options". Vous devriez voir apparaître le menu des options ci-dessous :



Cliquez sur "Add" pour ajouter un nouveau serveur IRC, qui sera en fait notre machine



Dans "Description", vous pouvez mettre ce que vous voulez, ça n'a pas d'importance. Dans "IRC Server" tapez "127.0.0.1" et dans "Port(s)" indiquez "8000". Et cliquez sur "Add". Voilà, notre mIRC est prêt à utiliser Httport.

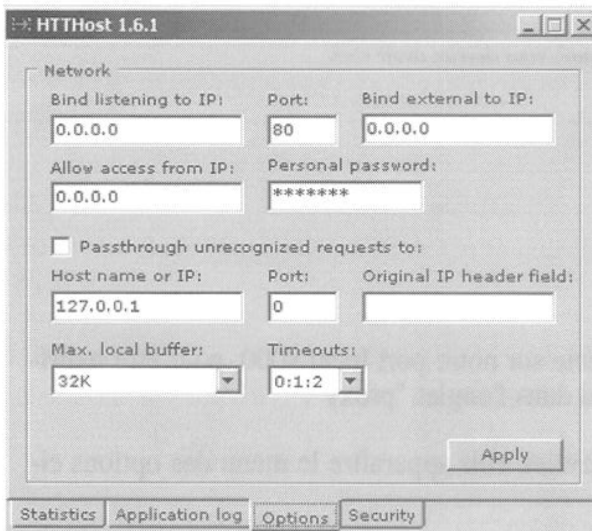
Voyons à présent si le proxy accepte le mode "connect", pour cela sur Httport , onglet "proxy" cliquez sur "Start". Prenez votre client IRC, sélectionnez le serveur que l'on vient de définir et cliquez sur l'éclair pour vous connecter. Si ça passe, votre proxy accepte le mode connect ! Prévenez l'administrateur réseau si pour que la configuration du proxy soit corrigée. Si cela ne marche pas, il est toujours possible d'utiliser une machine extérieure au réseau (la machine de chez vous par exemple).

```
*** Unable to connect (Connection refused)
-
*** Cancelled connect
```

Mon proxy n'accepte pas le mode connect, il va falloir utiliser Htthost.

Pour passer au travers du proxy, il va donc falloir installer Httphost sur une machine extérieure au réseau privé. Htthost jouera le rôle de relai entre le proxy et le serveur IRC. Il encapsulera les données pour faire croire au Proxy que les réponses du serveur IRC sont des réponses à des requêtes HTTP. Passons à la config de Htthost qui, je vous le rappelle, doit être installée sur une machine extérieure au réseau qui a un accès limité à Internet.

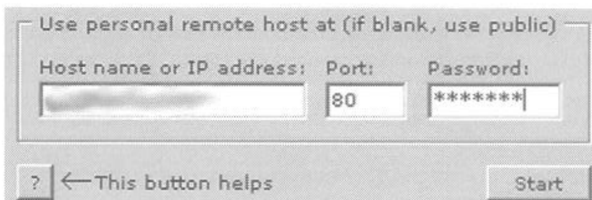
Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**



Dans l'onglet "Options" vous pouvez définir des restrictions. En mettant "0.0.0.0" comme IP, vous permettez à tout le monde de se connecter sur votre Htthost. Vous pouvez définir un pass d'access dans "Personal password". Ne tenez pas compte de l'option "Passthrough unrecognized request to:" et laissez-la décochée. Cliquez sur "Apply" pour appliquer les dernières modifications (ajout de pass par exemple). Il est important de laisser comme port le port 80, puisque c'est le port HTTP qui est a priori le seul sur lequel on puisse se connecter via le proxy du réseau protégé.

Votre Htthost est fin prêt.

Il faut maintenant indiquer à Httport que l'on désire se connecter via une machine extérieure qui utilise Htthost. Pour cela, retournez sur l'onglet "proxy" de Httport.



Dans "Host name or IP adress", mettez l'IP de la machine extérieure à l'entreprise où vous avez installé Htthost. Dans "port" spécifiez le 80, et dans "password", utilisez celui que vous avez défini dans Htthost

Cliquez sur "Start" pour activer Httport et reconnectez-vous sur le serveur IRC (127.0.0.1:8000) via mIRC. Si vous arrivez à vous connecter, c'est que n'importe quel utilisateur de votre réseau, ou n'importe quel virus, pourrait en faire de même ! La protection totale contre ce type d'usage malicieux des proxies est malheureusement impossible. Vous pouvezreprendre cet exemple pratique en sniffant votre réseau pour détecter les types de paquets envoyés, et écrire un filtre pour le sniffer qui vous avertira si quelqu'un essaie de faire du tunneling. N'oubliez pas aussi de configurer votre proxy pour qu'il n'autorise le relai HTTPS (SSL via la directive CONNECT) uniquement sur le port 443 (https). Ceci dit, on a parfois besoin de faire du tunneling pour des usages parfaitement légaux, mais prenez soin d'obtenir l'accord de l'administrateur.

III - Etude de paquets

Nous allons maintenant faire une étude de paquets afin de mieux comprendre le tunneling et les protocoles mis en jeu. Les captures ont été effectuées avec Ethereal. Commençons par une connection normale vers un serveur IRC, donc sans passer par un proxy.

Première capture :

- Les 3 premiers paquets représentent la négociation en trois fois du protocole TCP.
- Le 4 ème paquet correspond à la première commande IRC envoyée par notre client "NICK yopme" qui indique au serveur IRC le nick que vous avez choisi dans les options IRC. Vous pouvez remarquer que Ethereal détecte automatiquement le protocole IRC qui se trouve au dessus de la couche TCP.
- Le 5ème paquet indique que le serveur IRC fait une demande de connection (paquet TCP avec le flag SYN à 1) sur mon ordinateur (port 1080) ! En fait, il scanne mon port 1080 (syn scanning) pour vérifier que je ne passe pas par un proxy sock. En effet, si tel était le cas, le port du proxy serait ouvert et le serveur IRC nous déconnecterait. La plupart des serveurs IRC n'acceptent pas les connections via proxy.
- Le 6ème paquet signifie que mon PC répond au serveur avec un paquet TCP avec RST et ACK à 1. Si mon port avait été ouvert, mon PC aurait répondu par un paquet TCP avec ACK et SYN à 1 (il aurait accepté la connection sur le port 1080). Dans ce cas-là, le serveur m'aurait déconnecté.

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

LES COURS PAR CORRESPONDANCE DE THE HACKADEMY SCHOOL

```
-Global- [Logon News - Apr 30 2002] Si vous voyez une
  connection sur les ports 23, 1080, 3128, 8000, 8001,
  8080, or 8081 de 206.113.59.37, ce n'est pas une
  attaque mais notre detecteur de proxy. Regardez notre
  site web, ou nos motd, pour plus d'informations.
-
-Global- [Logon News - May 04 2002] En vous connectant
  sur ce reseau IRC, vous approuvez etre en accord, pour
  des questions de securité, pour vous memes, et les
  autres utilisateurs, avec la detection de proxy. (cf
  news sur la liste des ports testés.) Allez sur notre
  site web, pour plus d'informations. Kernel-IRC Staff
```

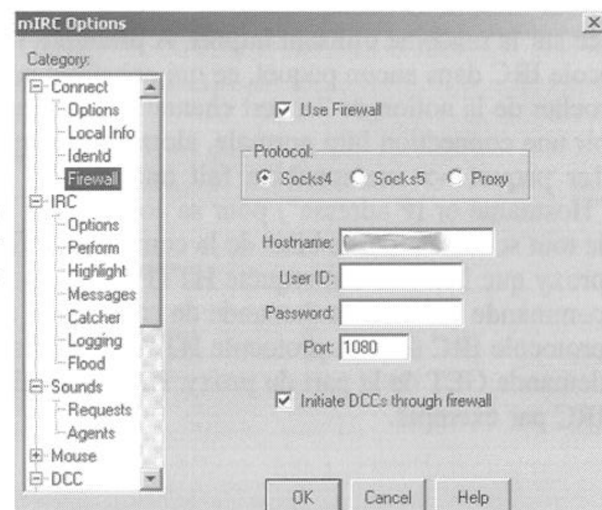
Le serveur IRC nous indique qu'il va scanner nos ports. Si vous jettez un coup d'œil dans la fenêtre status de votre mIRC lorsque vous vous connectez.

- 7ème paquet --> le serveur nous envoie un message IRC "NOTICE AUTH :*** Looking up your hostname..."
- 8ème paquet --> mon PC envoie une requête IRC au serveur pour s'identifier "USER yopyop "weed" "irc.kernel-irc.net" :Yopyop"
- 9ème paquet --> Tiens, tiens... Encore une demande de connection sur mon ordinateur de la part du serveur IRC, mais cette fois sur le port 113.
- 10ème paquet --> mon PC renvoie un ACK/RST car le port 113 est fermé.
- 11ème paquet --> le serveur acquiesce.
- 12ème paquet --> le serveur envoie trois messages IRC : "NOTICE AUTH :*** Found your hostname (cached)
  BeOS.Kernel-IRC.Net NOTICE AUTH :*** Checking ident...
  BeOS.Kernel-IRC.Net NOTICE AUTH :*** Checking for open socks server..." Il m'indique en dernier message qu'il vérifie sur mon PC la présence de ports ouverts (proxy). Même si c'est déjà fait ;)
- 13ème paquet --> ma machine accuse réception du paquet précédent
- 14ème paquet --> le serveur me renvoie 2 nouveaux messages IRC :
  "BeOS.Kernel-IRC.Net NOTICE AUTH :*** No ident response; username prefixed with ~
  BeOS.Kernel-IRC.Net NOTICE AUTH :*** No socks server found (good!)..."
  Le serveur a fini de scanner mon PC et m'indique qu'il n'a pas trouvé de proxy, donc pas de déconnection.
- 15ème paquet --> mon PC acquiesce.

On arrêtera là notre étude de paquets dans le cadre de la connection IRC sans proxy. A noter que lorsque l'on est connecté sur IRC, les serveurs utilisent le ping pour vérifier si l'on est bien connecté. Cela va se voir sur votre capture : le serveur envoie un "PING :2F60A6" et ma machine répond par un "PONG :2F60A6". Si jamais ma machine ne renvoie pas de PONG, alors le serveur coupe la connection.

Deuxième capture :

Nous allons maintenant étudier une connection IRC via proxy Socket. Nous différencierons ce cas-ci de celui basé sur le tunneling que nous aborderons par la suite. Voyons comment va réagir un serveur IRC face à un serveur proxy. Pour spécifier à mIRC de passer par un proxy SOCK, il vous suffit de lui indiquer un proxy dans la catégorie "Firewall" des Options. N'oubliez toutefois pas de mettre un serveur différent dans la catégorie "connect" (toujours dans les options).



Cochez "Use Firewall" pour demander à mirc de se connecter via un proxy. "Protocol" : choisissez socks4. "Hostname" : mettez l'IP du proxy SOCK que vous désirez utiliser. Et dans port laissez 1080. "Initiate DCCs throught firewall" n'a pas d'importance...

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

Retournez dans la catégorie "Connect" et vérifiez que vous avez bien mis un serveur IRC et non pas 127.0.0.1 qui était utilisé pour Httport.

- 1er puis 2ème paquets --> Requête et réponse DNS pour l'IP du serveur IRC (on peut remarquer que le serveur DNS nous donne plusieurs IPs)
- 3ème paquet --> Requête ARP de mon PC vers tout le réseau local (dont mon fournisseur d'acces ADSL) pour avoir l'adresse MAC de l'IP correspondant au serveur ProxySock.
- 4eme paquet --> La passerelle ADSL m'indique une adresse MAC pour ma requête de l'adresse IP du proxy socks. Note: ce n'est pas l'adresse MAC du proxy, attention ! Il s'agit de l'adresse MAC de la passerelle de mon fournisseur d'accès: la carte réseau qui se trouve au bout de ma ligne xDSL. Le protocole ARP ne peut pas être routé, il gère la transmission des données entre deux machines ou routeurs.
- 5ème, 6ème, 7èmes paquets --> Etablissement d'une connection TCP en 3 fois avec le serveur Proxy Sock
- 8ème paquet --> Utilisation du protocole Socks pour établir une connection avec le serveur IRC à partir du serveur Proxy Sock.

```
⊟ Socks Protocol
    Version: 4
    Command: 1 (Connect)
    Remote Port: 6667
    Remote Address: irc.kernel-irc.net (63.145.151.184)
    User Name:
```

- 9ème paquet --> le Proxy Sock acquiesce.
- 10ème paquet --> le bit PUSH est à 1 donc les données tcp sont directement envoyées au client IRC. Le proxy nous annonce que la connection a bien été établie avec le serveur IRC
- 11ème paquet --> mon PC envoie une requête (NICK Yop) au serveur IRC via le proxy sock.
- 12ème-42ème --> établissement de la connection avec le serveur IRC comme lors de l'étude précédente.
- 43ème --> le serveur IRC nous informe qu'il va scanner nos ports
- 66ème --> le serveur IRC a détecté que je passais par un serveur Proxy, alors il nous informe de cette découverte...
- 67ème --> le serveur IRC envoie un paquet TCP avec le flag FIN à 1, ce qui a pour but de couper la connection. Etant donné que leur serveur a scanné le proxy, et par conséquent le port 1080, il a vu que le port était ouvert, il en déduit donc que je passe par un proxy Sock. (bien vu ;).

On en conclut que la connection via proxy Sock ne passe pas, car le serveur IRC la détecte. Au niveau de la connection avec le serveur IRC via Proxy, elle est presque identique à la première capture, à la différence près que l'on utilise le protocole sock pour envoyer nos données vers le serveur IRC. Il faut bien voir que l'on ne communique pas directement avec le serveur IRC mais que c'est bien le serveur Proxy qui envoie les données vers le serveur IRC.

Troisième capture :

Pour finir, nous allons étudier la connexion à IRC en utilisant Httport et Htthost. La capture a été effectuée sur la machine utilisant httport. A première vue, vous pouvez remarquer qu'Ethereal ne détecte le protocole IRC dans aucun paquet, ce qui constitue une grande force du tunneling pour les pirates. C'est à rapprocher de la notion de “covert channel”: un administrateur qui regarderait de loin le traffic réseau croirait voir une connection http normale, alors qu'il s'agit en fait d'un chat irc.

- 1er paquet --> ma machine fait une requête au proxy (celui qui est dans l'onglet "proxy", champ "Hostname or IP adresse") pour se connecter à la machine sur laquelle est installé htthost (80.11.xx.xx), le tout se faisant par le biais de la commande "GET http://80.11.xx.xx/...". Cette commande fait croire au proxy que l'on fait une requête HTTP via la commande GET. Ce qui se trouve à la suite de l'URL de la commande GET est la demande de connection au serveur IRC, mais ceci est masqué (on a encapsulé le protocole IRC dans le protocole HTTP !). Htthost est là pour décoder la suite de l'URL lorqu'il reçoit la demande GET de la part du proxy. Htthost va alors générer une connection permanente, avec un serveur IRC par exemple.

© 2001-2002-2003 **DMP**

IV - L'altération de données réseaux

1 - Principes et fondements.

Dans ce chapitre, nous allons aborder le principe théorique et nous en ferons ensuite une démonstration pratique. Tout ici est très simple à comprendre. Nous espérons vous faire prendre conscience, par cette étude, de l'importance d'adopter des protocoles de communication non seulement chiffrés, mais aussi utilisant un mécanisme d'authentification par certificats (comme le permet SSL). Sans quoi vos données resteront lisibles et modifiables par un pirate grâce à l'attaque dite de "l'homme au milieu".

Le principe de l'altération de données réseaux s'effectue sur la base d'un sniffing. Les données capturées par une machine quelconque ne sont cependant pas ici capturées passivement. Pour pratiquer une altération de données, il faut être sur une voie de communication réseau entre deux machines, ou davantage, afin de pouvoir non seulement capturer les données, mais les modifier et les retransmettre.

Exemple théorique :

Pour communiquer de la machine A à la machine B, les données transitent par la machine Relai. C'est sur la machine Relai que va porter l'attaque.



Les réponses de la machine B à la machine A ne doivent pas obligatoirement transiter par la machine Relai. Par exemple, A a pu demander à B de contacter une machine distante autre que Relai, disons "Betty". Mais cette demande, une fois altérée par Relai, peut transformer la demande de connection vers "Betty" par une demande de connection vers "Harry", machine d'un complice.

Vous n'y croyez pas ? Vous voyez mal comment il est possible de mettre en place une telle attaque ? Alors passons directement à l'étude pratique.

2 - Application pratique : le cas d'un serveur proxy.

Le proxy HTTP constitue un très bon exemple pour une approche pratique démontrant cette technique d'attaque. En effet, le proxy est une machine qui va servir de relai entre un client et un serveur, généralement un serveur HTTP. Les proxies sont souvent utilisés pour "anonymiser" des connexions sur Internet entre un internaute et un site Web, car sur les paquets qu'ils relaient entre eux et le serveur HTTP, ce sont leurs adresses IP qui sont utilisées. Le hic est qu'en utilisant un proxy dont vous n'avez pas le contrôle, il est impossible de savoir si le trafic n'y est pas surveillé, voire même altéré. Certes, tout est une question de confiance, et de façon empirique, il est peu probable que ce soit le cas.

Pour notre exemple, nous allons mettre en place un serveur proxy un peu particulier qui a la capacité de capturer les données HTTP émises par le client et renvoyées par le serveur. Le logiciel va capturer les données HTTP, vous permettre de les modifier, et les relayer quand bon vous semblera. Nous allons ainsi voir, à travers quelques scénarios, en quoi ce type d'application peut être utile à un pirate. Notez qu'un véritable pirate devrait utiliser d'autres outils pour réaliser une attaque efficace, mais le principe serait exactement le même.

Première étape : installer le serveur proxy adéquat.

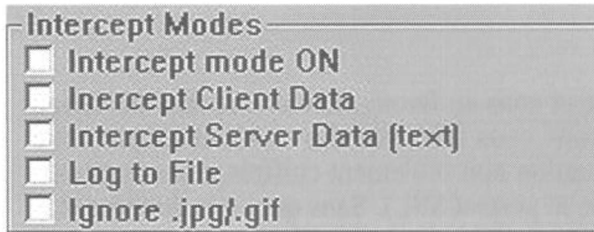
Rendez-vous sur le site http://www.digizen-security.com/downloads.html pour le téléchargement du logiciel Achilles. Prenez la version 0.27, la dernière. Il est possible qu'au moment de la lecture de ces lignes, une nouvelle version soit sortie, mais l'utilisation du logiciel, tant sur le fond, que beaucoup sur la forme, devrait rester sensiblement similaire.

Dans le fichier zip sont enfermés le logiciel et les fichiers qui lui sont liés. Extrayez-les dans un répertoire vide et exécutez directement Achilles.

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

Deuxième étape : configurer le serveur.

Le logiciel est très simplement configurable. La fenêtre de configuration se présente comme ceci :



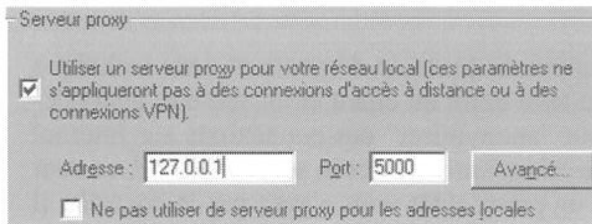
Pour lancer le mode de capture des trames HTTP, cochez "Intercept mode ON". Le serveur proxy fonctionne même dans le cas où vous n'auriez pas coché cette option. Les autres options servent respectivement à:

- Intercept Client Data : intercepter (bloquer et mettre en attente) les données envoyées par le client (la personne qui tente de joindre un serveur HTTP, comme un site).
- Intercept Server Data : intercepter les données transmises par le serveur HTTP (généralement les réponses d'un site à une requête).
- Log to File : enregistrer les résultats dans un fichier.
- Ignore .jpg/.gif : ignorer les requêtes de téléchargement des images sur les pages web (recommandé !).

Troisième étape : configurer le client.

Cette étape implique de pouvoir faire modifier au client, que ce soit par vous-même ou un intermédiaire, ses paramètres de connections aux serveurs HTTP. Le logiciel client utilisé généralement pour ça sur Windows est Internet Explorer. Nous allons donc voir comment configurer IE afin de l'obliger à communiquer avec un proxy pour contacter un site Web. Même si cela paraît simple, c'est nécessaire.

1. Ouvrez une fenêtre Internet Explorer ;
2. Allez sur l'onglet "Outils" ;
3. Allez sur Options Internet ;
4. Cliquez alors sur "Connexions" ;
5. Cliquez sur "Paramètres réseau" :
6. Cochez "Utiliser un serveur proxy (...)" ;



7. Remplissez la case Adresse par votre localhost (adresse IP : 127.0.0.1) et indiquez comme numéro de port 5000, ou, du moins, le même que celui utilisé par Achilles pour recevoir les données :



(fenêtre d'Achilles)

Par la configuration demandée, vous faites relayer vos paquets par vous-même pour contacter un site Web. Au niveau logiciel, c'est Achilles qui se charge de ce relais, en recevant les données par le port 5000. Ainsi, la requête part de IE sur votre machine, pour aller sur Achilles toujours sur votre machine, et Achilles se charge de relayer les trames de requêtes et de réponses entre IE et le serveur Web.

Voyons à présent divers scénarios dans lesquels nous allons mettre en pratique notre démonstration. Pour cela, vous devrez avoir paramétré IE comme vu plus haut, et faire tourner Achilles de différentes façons sur le plan de la configuration, façons que nous allons étudier maintenant.

© 2001-2002-2003 **DMP**

Pratique / Scénario 1 : Subtiliser des mots de passes saisis dans des pages Web.

Un utilisateur quelconque va sur un site Web et y saisit un mot de passe. Cet utilisateur a votre machine comme serveur proxy. Voyons comment un pirate, en utilisant un procédé similaire à celui-ci au niveau des routeurs Internet, pourrait dès lors subtiliser des mots de passe.

Configuration Achilles requise :

- Intercept mode ON : activé. <
- Inercept Client Data : activé <
- Intercept Server Data (text) : désactivé
- Log to File : désactivé
- Ignore .jpg/.gif : activé <

A pratiquer :

1. Allez sur une page Web appropriée pour faire saisir un mot de passe (http://www.caramail.com, par exemple), faites passer toutes les requêtes nécessaires envoyées par le navigateur pour qu'elles ne se bloquent pas au niveau d'Achilles. Faites-les transiter grâce à la touche Send d'Achilles. Ne les modifiez pas pour l'instant ;
2. Tapez un faux login et un faux mot de passe ;



3. Soumettez-les ;
4. Dans Achilles apparaît alors une requête POST dans laquelle sont contenus le login et le mot de passe de l'utilisateur.

```
Send | Find/Rep
POST /scripts/baltop HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Referer: http://www51.caramail.lycos.fr/general.jsp?N=16262&C=0&
Accept-Language: fr
Content-Type: application/x-www-form-urlencoded
Proxy-Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)
Host: www51.caramail.lycos.fr
Content-Length: 66
Pragma: no-cache
Cookie: LBC=AOKGKHAGGBBAFDJEADGGJNJGDBPMNKODBJLAEHKPOFFCBIPJ;
IDENTIFIANT=NCHHANINHNAEEIOIAIHFDEJGAKMGCGLGJCKAICNBGFCCJEJP;
LBC=d64be3b80f75c30c930a6f44fbb9fba9

LOGIN1=[illegible]&PASSWORD1=kezako&Enter=+Entrer+&TZ=200205240053
```

Explications :

La configuration d'Achilles est ici adaptée à capturer toutes les données émises par le client, et uniquement par le client, ce qui inclut l'envoi des logins et des mots de passe dans le cas de figure présenté.

Pratique / Scénario 2 : rediriger l'utilisateur vers un autre site web que celui appelé

Il faut savoir que cette pratique peut avoir beaucoup d'utilité pour un pirate. En effet, supposons qu'un utilisateur lambda souhaite accéder à ses services financiers par Internet. S'il saisit ses mots de passe sur des services sécurisés qui chiffrent les données, un pirate pourrait lui faire croire qu'il saisit ses logins et mots de passe sur ce même service après l'avoir redirigé vers un site à lui, car l'allure de la page serait identique, et surtout parce que l'utilisateur a tapé la bonne adresse dans son navigateur, donc il est en toute confiance ! Méfiance donc; vérifiez toujours la validité des certificats SSL des sites que vous visitez.

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

Configuration Achilles requise :

- Intercept mode ON : activé. <
- Inercept Client Data : désactivé
- Intercept Server Data (text) : activé <
- Log to File : désactivé
- Ignore .jpg/.gif : activé <

En pratique :

Nous allons juste vérifier qu'il peut être possible pour un pirate de rediriger l'utilisateur, alors que celui-ci se rend sur une page web de "confiance". Pour cela, laissez toujours Achilles ouvert et actif selon la configuration vue plus haut, et IE configuré pour l'utiliser comme relai.

1. Dans IE rentrez l'adresse http://www.google.fr ;
2. Dans Achilles apparaît alors le contenu de la page HTML de Google ;
3. Vous pouvez alors altérer le contenu de la page et y insérer un script tel que celui-ci : <script>window.location="http://www.dmpfrance.com"</script>, entre deux balises HTML.
4. L'utilisateur se verra alors redirigé vers http://www.dmpfrance.com.

Vous pouvez aussi créer votre propre contenu à injecter à Achilles, un contenu préparé qui n'attend que d'être vu par l'utilisateur.

1. Allez sur http://www.google.fr ;
2. Dans Achilles, effacez l'ensemble du contenu de l'encadré où s'affichent les données ;
3. Ecrivez uniquement le script <script>window.location="http://www.dmpfrance.com"</script>
4. Envoyez-les (Send) ;
5. L'utilisateur se retrouve ainsi redirigé. Mais à la place de <script>window.location="http://www.dmpfrance.com"</script> le pirate aurait pu rentrer le code HTML d'une page Web préparée à l'avance qui a l'allure du site que l'utilisateur s'attend à voir...

Au final, un pirate pourrait duper un utilisateur lambda sans éveiller ses soupçons. Que ce soit pour lui voler des cookies, le rediriger sur des pages ou lui en fabriquer, l'empêcher d'accéder à ses comptes ou lui subtiliser ses mots de passe, la condition sine qua non reste que l'utilisateur-victime doit utiliser un routeur ou un serveur proxy contrôlé par le pirate.

Pour vous protéger un peu si vous n'utilisez pas SSL, évitez d'utiliser un proxy, en particulier si vous ne le contrôlez pas, comme celui de votre founisseur d'accès. Malheureusement, cela n'est pas toujours possible car certains fournisseurs vous obligent à utiliser un proxy dit "transparent".

V - L'IP Spoofing

Ce chapitre risque, soyons honnête, et sans vouloir vous martyriser, de vous faire assez mal à la tête. Pour certains qui n'ont aucune connaissance en réseaux, accrochez-vous. Pour ceux qui possèdent deux ou trois notions de TCP/IP et maîtrisent un peu le sniffing, ce chapitre ne devrait pas (trop) les torturer. Sauter ce chapitre n'entamera pas vos requis pour la suite du cours (sauf en ce qui concerne l'idle scanning), mais il vous aidera, une fois le sujet maîtrisé, à mieux aborder les communications réseaux, le spoofing en lui-même (comme Kevin Mitnick) et les protocoles réseaux. Vous verrez toutefois qu'en pratique, ce n'est pas aussi sorcier que ça en a l'air...

1 - Définition

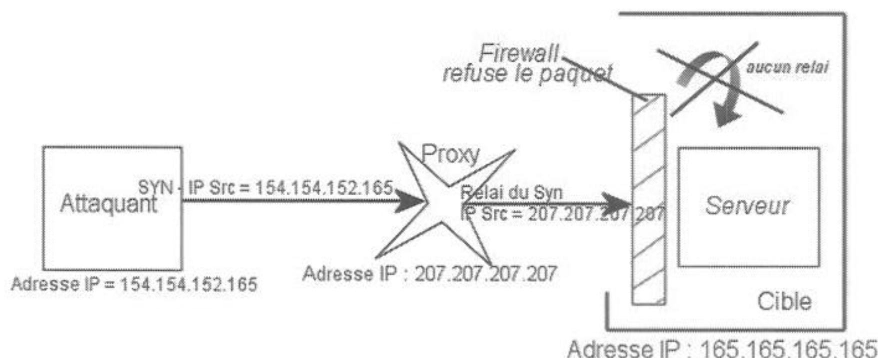
Pour définir clairement ce qu'est le spoofing, abolissons déjà les mauvaises idées reçues. Ainsi, il faut se montrer ferme en ce qui concerne le "spoofing par des proxies". Même si un proxy peut servir de relais à certains services (comme le service HTTP), il n'empêche absolument pas quelqu'un de vous retrouver. Et dans le cadre d'une enquête judiciaire, le proxy n'est absolument pas une protection à une éventuelle remontée aux sources. Il permet peut-être de tromper d'éventuels logs, mais même d'un point de vue technique, il ne permet pas de véritables "attaques par spoofing".

© 2001-2002-2003 **DMP**

Le spoofing est une technique de modification de l'adresse IP au niveau des paquets réseaux, qui va permettre l'accès à des serveurs, en théorie protégés par des systèmes de sécurité filtrant les paquets lancés vers une cible (un ordinateur d'un réseau, en général). Le spoofing peut donc servir à des attaques, et l'utilisation de proxies constitue une manière simple de "spoofer" puisque la modification de l'adresse IP ne s'effectue jamais véritablement que lors du relais, et pour prendre celle du proxy.

Ce qui veut dire que le spoofing permet de passer à travers des protections qui se basent sur un filtrage de l'adresse IP. Sur un plan théorique, cela semble simple. En spoofant, vous ne changez pas véritablement votre adresse IP. En revanche, les paquets que vous transmettez semblent provenir d'une autre source.

Schéma explicatif



L'IP de l'attaquant ne semble modifiée qu'à partir du moment où ses paquets passent à travers le proxy. Dans un système de sécurité se basant sur un filtrage IP, passer à travers un proxy ne sert à rien (à moins que l'adresse IP du proxy soit autorisée par le système). De plus, les paquets de l'attaquant passent à travers d'autres systèmes de relai avant d'arriver au proxy. C'est d'autant plus de chances de se faire repérer.

2 - Approfondissement théorique (nécessaire) :

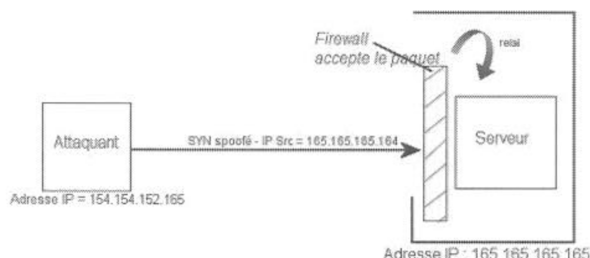
Lorsque quelqu'un crée une véritable attaque par spoofing, les paquets sortants sont déjà modifiés au niveau de l'en-tête IP, qui contient un champ consacré à l'adresse IP source, celle de l'émetteur. Voyons comment se construisent les en-têtes IP de paquets de données.

En-tête IP

0 — 16 — 32 bits

Ver.	LET	Type de service	Longueur totale	
Identification			Flags	Fragment Offset
Durée de vie		Protocole	Checksum d'en-tête	
Adresse source				
Adresse destination				
Option + Bourrage				
Data				

Les modifications se font au niveau de "l'adresse source". Notez que l'adresse IP que vous fournit votre provider ne change pas, ce sont des paquets fabriqués et truqués qui sont émis. Analysons à travers un schéma ce que fait une attaque par spoofing.



Voilà en gros comment se présente un spoof simple réussi. Seulement, il y a plusieurs étapes qui succèdent à cette première action. Il faut, pour cela, savoir que toute connection TCP vers un serveur s'établit sur le modèle bien précis : requête de connection, réponse du serveur qui confirme la requête et demande au client

Une publication The Hackademy School
© 2001-2002-2003 DMP

l'autorisation d'une connection, puisque le client répond en autorisant à son tour le serveur à établir la connection. D'où le célèbre schéma

client --> SYN --> serveur
client <-- ACK/SYN <-- serveur
client --> ACK --> serveur

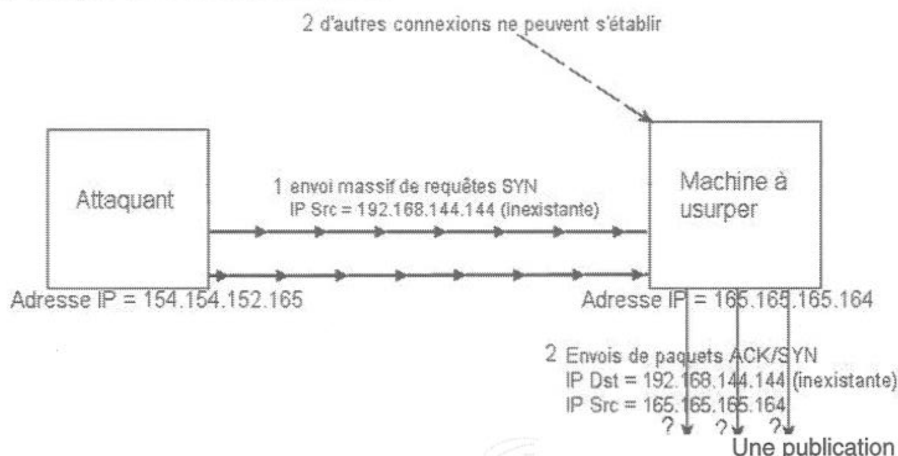
dont vous avez entendu parler dans le cours précédent, qui concerne la manière dont s'établissent les connections via TCP (Transfer Control Protocol). Dans ce schéma, SYN signifie "synchronisation" et ACK "acknowledgement". Le processus permet très vite de comprendre quel problème se pose à l'attaquant. Le serveur va répondre à la machine dont l'adresse a été spoofée. Celle-ci, recevant un paquet SYN/ACK inattendu de la part du serveur, va y répondre par un paquet TCP dont le bit RST (ReSeT) (on appelle ça aussi un "flag", donc on parle de "flag RST" dans le cas présent) est activé, mettant fin à la tentative de connection. Voici comment se construisent les en-têtes TCP :

U = URG, Pointeur de données urgentes significatif
A = ACK, Accusé de réception significatif
P = PSH, Fonction Push
R = RST, Réinitialisation de la connexion
S = SYN, Synchronisation des numéros de séquence
F = FIN, Fin de transmission

Une attaque par spoofing sera donc ratée à cause du RST renvoyé par la machine usurpée. Le problème est donc que la machine spoofée a été contactée pour l'établissement d'une connection, et donc l'attaque est un échec. La machine spoofée n'ayant jamais demandé de connection, elle envoie un paquet d'annulation (RST). Là se pose le premier problème : l'attaquant est en fait "aveugle" (on appelle ça tout simplement : "a blind attack", une "attaque en aveugle"), car il n'y a jamais de connection qui s'établit vers lui. Pour résoudre ce premier problème (il y en a un autre que nous verrons par la suite), l'attaquant doit d'abord invalider la machine pour laquelle il se fait passer. Voyons comment les pirates effectuent un processus d'invalidation.

Lors de l'établissement d'une connection client/serveur, le serveur va envoyer un paquet SYN/ACK, comme vu plus haut sur le schéma, et va attendre la réponse du client. Tant qu'il n'y a pas de réponse à ce paquet SYN/ACK, le serveur va attendre la réponse, et la connexion ne peut être établie.

Il existe une limite aux nombres de requêtes SYN qui peuvent être envoyées sur un même socket (couche logicielle qui gère les transmissions réseaux). C'est le "backlog", il représente la longueur de la file d'attente des transmissions incomplètes. Dans le cas où cette limite est atteinte, les futurs paquets de connection sont ignorés. Là se porte tout l'intérêt de la chose. Il suffit de flooder avec des paquets SYN une machine à distance, pour lui faire ignorer les futurs paquets qu'elle recevra. Le flood doit se faire avec une IP modifiée dans l'en-tête des paquets, avec une IP qui n'existe pas, de sorte que le flood soit valide. Autrement, la cible floodée recevrait des paquets RST de l'hôte spoofé lors de l'attaque. Voilà comment se présente une attaque d'invalidation réussie.



Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

Grâce à l'invalidation de la machine, l'attaquant va pouvoir se faire pleinement passer pour la machine invalidée. Attention toutefois au "timeout" sur les connections client/serveur : la machine floodée arrête au bout d'un certain temps la gestion de la tentative de connection en cours pour passer à la suivante. De plus, si un reboot manuel de la machine est fait, si un administrateur s'aperçoit du problème par exemple, alors le flood aura été un échec.

Afin de bien résoudre ce problème de "l'attaque à l'aveugle", le pirate peut, après avoir invalidé la machine pour laquelle il se fait passer, utiliser le "source routing". Cette option s'effectue dans le champ "option" de l'en-tête IP, et va permettre de spécifier une route de retour au paquet envoyé. Ainsi, grâce à un peu de sniffing, l'attaquant peut lire le contenu des trames de retour. Le source routing a aussi une application tout aussi intéressante, que nous verrons bientôt. Attention toutefois, il est utile de préciser que la plupart des routeurs actuels (99% des routeurs de la planète) ne prennent plus en compte cette option. Pour ne pas avoir à faire du blind spoofing, d'autres méthodes que nous détaillerons plus tard peuvent être envisagées.

Entre-temps, l'attaquant doit se faire passer pour la machine invalidée, qui a une relation de confiance au sein du système de filtrage, et ce n'est pas gagné. C'est le deuxième problème majeur d'une attaque par spoofing, où n'est pas pratiqué le source routing. Bien que la machine soit invalidée, que son adresse IP soit parfaitement spoofée, le pirate doit en plus faire une opération de prédiction des "numéros de séquences".

Le numéro de séquence identifie la place du premier octet de ce segment dans le flux de données provenant de l'émetteur. Ce numéro est un nombre non-signé sur 32 bits qui retourne à 0 après avoir atteint $2^{32}-1$. Lorsqu'une nouvelle connexion est en train de s'établir, le flag SYN est mis à 1 (mettre à 1 signifie que le flag SYN est actif, "on"). Le champ numéro de séquence contient le numéro de séquence initial (ISN) choisi par la machine pour une connexion. Le numéro d'acquittement (Acknowledgment number) contient le numéro de séquence suivant, que l'émetteur de l'acquittement s'attend à recevoir. C'est par conséquent le numéro de séquence du dernier octet reçu avec succès + 1. Une connexion TCP/IP va donc se présenter ainsi :

Etablissement d'une connection sous TCP/IP

Ordre d'envoi des paquets	Direction	Valeurs pour les numéros de séquence	Valeurs pour les flags SYN et ACK
Paquet 1	Client à Serveur	(ISN) Seq = x	SYN = 1 ACK = 0
Paquet 2	Serveur à client (réponse)	(ISN') Seq = y ACK = x + 1	SYN = 1 ACK = 1
Paquet 3	Client à serveur	Seq = x + 1 ACK = y + 1	SYN = 0 ACK = 1

ce qui se résume à :

client --> (ISN) Seq = x | flag SYN --> serveur
client <-- (ISN') Seq = y et ACK num = x + 1 | flag SYN/ACK <-- serveur
client --> Seq = x + 1 et ACK num = y + 1 | flag ACK --> serveur

NB : Ne pas confondre Acknowledgement number (Ack num) et ACK qui est le flag associé.

L'attaquant doit avoir une idée du nombre contenu dans le numéro de séquence initial de la cible. Ceci peut se faire par sniffing, par exemple, si le source routing est effectif. Il faut savoir que les vieilles données de changement de l'ISN (128 000/seconde et 64 000/connexion) ne sont plus valables sur de nombreux systèmes informatiques qui disposent de générateurs de nombres aléatoires performants. Cependant, de nombreux périphériques réseau (y compris des imprimantes !) ont encore un ISN prévisible. Lorsque l'attaque commence réellement (car il s'agissait là de la dernière étape à une attaque par IP Spoofing), plusieurs possibilités sont à prévoir. Le numéro d'ACK correspond parfaitement, et, dans ce cas, les données sont placées en attente d'être traitées. Si le numéro d'ACK est inférieur au numéro attendu, alors le paquet est supprimé. Si le numéro est supérieur à ce qui est attendu mais reste dans la limite d'un faible écart, il est placé en attente comme paquet intermédiaire. Si le numéro est vraiment trop supérieur, le paquet sera supprimé.

Notez que nous parlons ici de spoofing IP sur le protocole TCP. Le spoofing IP est trivial à réaliser sur le protocole UDP (car il n'y a pas besoin de deviner un numéro de séquence). Préférez donc TCP !

© 2001-2002-2003 **DMP**

3 - Passer à la pratique :

Pour monter une attaque de type spoofing chez soi, il est nécessaire, pour faire au plus simple, d'être dans un environnement réseau local adapté (équipé d'un HUB par exemple) et de bénéficier du contrôle d'au moins 2 machines. Nous verrons ici comment monter une attaque de type spoofing sur l'environnement suivant :

- Une machine sous Windows 98 ou Windows 2000 (ou autre NT) ;
- Une machine sous un système NT ;
- Les deux machines reliées en réseau local, capables de communiquer entre elles, à travers un HUB.

J'ai choisi cette configuration car elle est courante chez nombre de personnes et s'adapte à l'environnement logiciel qu'il va falloir mettre en place pour monter notre simulation d'attaque. Ainsi, vous aurez besoin de :

- Un serveur qui accepte des connexions (nous verrons comment installer et configurer netcat à cet effet) ;
- Un "packet forger" (outil de construction de paquets), tel Winject. D'ailleurs, étudions-le tout de suite.
- Ethereal sur le poste où tourne Winject (votre station NT obligatoire), prêt à l'emploi ;

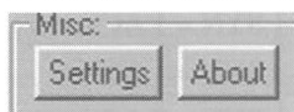
Winject : Configuration anté-utilisation.

Téléchargez tout d'abord Winject v0.96b sur http://big.badlink.net, dans la section "Download". Le fichier zip contient Winject et les fichiers qui lui sont attachés, prêt à l'emploi. Décompressez l'ensemble des données dans un répertoire adapté, créé pour l'occasion par exemple. Lancez le logiciel Winject.exe.

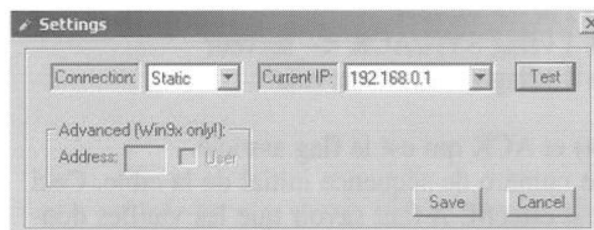
1. Une fenêtre de bienvenue s'ouvre. Validez ;



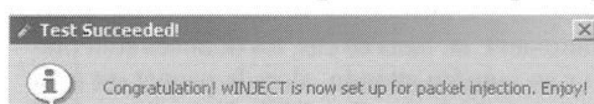
2. Passez directement à la configuration nécessaire du logiciel, en cliquant sur "Settings" dans l'encadré "Misc:" ;



3. Sur la case "Connection" choisissez "Static", et pour "Current IP", choisissez l'adresse IP qui vous est attribuée au sein du réseau local, commençant généralement par 192.168. ;



1. Cliquez sur Test. Le Test devrait parfaitement fonctionner. S'il y a un problème à ce niveau, c'est que vous avez fait une erreur ou que vous n'avez pas respecté les critères de configuration nécessaires à cet exercice.



1. Cliquez sur Save ;
2. Vous voilà revenu à la fenêtre principale. Laissez le logiciel en stand-by, le temps pour nous de procéder à l'installation d'un serveur.

© 2001-2002-2003 **DMP**

Netcat : installer et configurer Netcat.

Netcat est un peu comparable à une boîte à outil fonctionnant en ligne de commandes. Il est gratuit, à disposition pour différents OS, y compris Windows, comme nous allons le voir. Il intègre un mode "écoute" qui lui permet de monopoliser un port en attente d'une connection. Cela permet normalement de recevoir des données textes, voire même des fichiers. Nous allons ici l'utiliser pour lui faire accepter une connection avec un ordinateur qui n'existe pas, sur le port qu'il met en écoute.

Vous trouverez Netcat en téléchargement sur http://www.atstake.com/research/tools, dans la section "Network Utility Tools". Là encore, le fichier zip contient netcat prêt à être installé. Nous vous conseillons de décompresser netcat dans votre répertoire COMMAND qui se situe dans votre répertoire Windows par défaut (C:\Windows\Command\). Ainsi, depuis une fenêtre MS-Dos, vous pourrez accéder au logiciel en tapant simplement "nc". Pour la suite du cours, nous ne verrons l'appel à l'application nc.exe que sous la forme "nc". Maintenant, procédons aux manipulations adéquates qui vont vous permettre de mettre en écoute Netcat sur un port, disons le port 21.

1. Ouvrez une fenêtre MS-Dos ;
2. Tapez, sous Windows 98 ou votre deuxième système NT, les commandes suivantes :
   "nc -L -p 21" ou "nc -L -v -v -p 21" pour avoir plus d'informations sur l'activité de netcat pendant son exécution ;
3. Netcat passe alors en mode écoute. Il acceptera toute tentative de connexion sur le port 21 de la machine sur laquelle il est actif.

A ce stade du cours, vous devriez avoir :

- Une machine A (sous Windows 98 ou votre deuxième NT) qui fait tourner netcat en mode écoute sur le port 21.
- Une machine B (sous 2000 ou autre NT), qui fait tourner Ethereal non lancé, et Winject encore à l'état de post-configuration.
- Les deux machines sur le même réseau local, aptes à communiquer.

Visualisez bien vos noms de machines (A et B) car tout au long des explications à suivre, nous ne les appellerons que "machine A" et "machine B" ou encore "A" et "B".

La mise en pratique - étape 1 : analyser l'objectif à atteindre.

Si vous pensez qu'il est inutile de s'embêter à analyser les processus courants pour mieux comprendre comment se fait l'attaque par spoofing et souhaitez plutôt aller vers sa mise en place (sans même une petite préparation ?), alors allez directement à l'étape 2.

1. Sur votre machine B, lancez Ethereal pour qu'il commence à capturer les données. Mettez-y un filtre TCP, de sorte qu'Ethereal ne capture que des paquets TCP (cf premier chapitre pour la manipulation des filtres). Ou plus simplement encore, dans le bas de la fenêtre principale d'Ethereal, dans la case consacrée au bouton "Filter", tapez "tcp" puis cliquez sur le bouton en question ;
2. Lancez une session de capture avec Ethereal.
3. Lancez une connexion TCP avec telnet sur votre machine B vers la machine A, sur son port 21. Donc, allez dans le menu Démarrer, puis Exécuter et entrez "telnet [adresse IP A] 21" où [adresse IP A] correspond à l'adresse IP de votre machine A.
4. Dès que la connexion est établie (telnet tourne, et il ne se passe rien), fermez la session de capture d'Ethereal.
5. Parmi les informations à l'écran, cherchez le paquet de données émis en TCP par la machine B vers la machine A. Cherchez en vous aidant des onglets, ou en faisant défiler les informations (la session de capture a été d'une courte durée).

```
3 3.969456 CLAD-SERVEUR   CLIENT    TCP   4853 > ftp [SYN] Seq=4293842175 Ack
4 3.969621 CLAD-CLIENT    SERVEUR   TCP   ftp > 4853 [SYN, ACK] Seq=25725366
5 3.970010 CLAD-SERVEUR   CLIENT    TCP   4853 > ftp [ACK] Seq=4293842176 Ack
```

6. Sur l'image ci-dessus, se trouve l'exemple d'une connexion établie en TCP entre une machine nommée SERVEUR (qui devrait être votre machine B) et une machine CLIENT (votre machine A).

© 2001-2002-2003 **DMP**

Ligne 1 : l'émission de la demande de connexion a été faite par la machine B grâce à un paquet TCP avec le flag SYN activé, émis depuis le port dynamique 4853 en direction du port 21 de la machine A. On constate ici que "ftp" est affiché, même si le protocole FTP n'est pas utilisé. Ethereal traduit en fait les services qui leurs sont généralement associés). Factuellement, c'est le fait d'avoir lancé une connection avec Telnet sur la machine B. [SYN = 1 ; ACK= 0 ; Seq = X]

Ligne 2 : la machine A répond à la demande de connection par un SYN/ACK, afin de faire sa propre demande et signaler son acquittement. Factuellement c'est l'application Netcat qui répond positivement sur la machine A.
[SYN = 1 ; ACK = 1 ; Seq = Y ; Ack Num = X+1]

Ligne 3 : la machine B parachève l'établissement de la connection par un paquet SYN, signe qu'il autorise lui aussi la connexion à s'établir de son côté.
[SYN = 0 ; ACK = 1 ; Seq = X+1 ; Ack num = Y+1]

Tout le principe va consister à forger deux requêtes (Ligne 1 et Ligne 3), l'une pour établir la fausse connexion, l'autre envoyée dans un court laps de temps qui va servir à valider la connexion, à la quasi-identique des paquets vus en exemple. L'attaque consiste à forger des paquets avec une IP Source inexistante, à en corriger les checksums (un petit champ dans l'en-tête TCP qui sert à vérifier l'intégrité des données), et à y placer les flags au bon endroit pour les envoyer au bon moment avec des numéros de séquences valides.

Nous obtiendrons l'accomplissement de ces objectifs en sachant que Winject va nous servir à forger les données, Ethereal à faire la correction du Checksum et à sniffer la requête SYN/ACK de retour de la machine A (Ligne 2) pour la capture du numéro ISN(numéros de séquence "seq=") de la machine A.

La mise en pratique - étape 2 : atteindre l'objectif.

Revenez au mode de travail initial :

- Machine A avec netcat actif sur le port 21 ;
- machine B avec Ethereal, prêt à l'emploi, mais non activé (les données utilisées dans l'exercicre pré cédent doivent-être effacées) et Winject.

La première étape va consister à forger les paquets de données de la façon la plus efficace possible. Nous allons donc forger avec Winject deux paquets :

- un paquet TCP avec le flag SYN et un numéro de séquence inventé et une IP source spoofée mais qui n'existe pas (donc incontactable), et un checksum à 0.

- un paquet TCP avec le flag ACK et un numéro de séquence qui est le numéro de séquence inventé + 1, une valeur pour ACK qui sera celle de l'ISN + 1 du paquet de retour de A, sniffé par Ethereal (n'oubliez pas que le HUB renvoye les paquets à l'ensemble du réseau), et un checksum à 0.

1. Prenez Winject ;
2. Cliquez sur le bouton "Open" de la case "Packet :" et allez dans le répertoire "PACKETS" du dossier où est installé Winject.



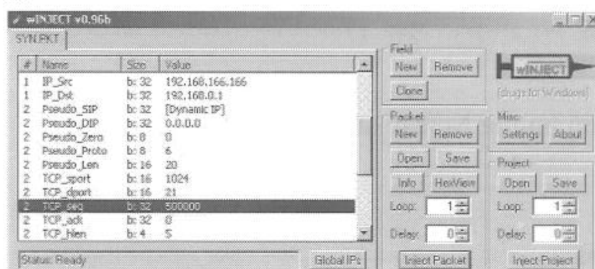
3. De là, ouvrez "SYN.pkt" ;
4. Winject permet alors d'éditer le paquet. Celui-ci est construit sur le modèle d'un paquet TCP avec un flag SYN. On est donc très proche des paquets que l'on souhaite forger. Il suffit d'apporter quelques modifications.

© 2001-2002-2003 **DMP**

5. D'abord, modifions l'IP Source et indiquons une adresse IP qui n'existe pas, une adressse IP de réseau local inexistante (par exemple 192.168.166.166, ou autre si votre réseau local relaie déjà, par le plus grand des hasards, une machine sous cette IP). A cet effet, double-cliquez sur la ligne "IP_Src". S'ouvre une fenêtre d'édit du champ du paquet.
   o Dans "Format :" rentrez "IP" ;
   o Dans "Value :" indiquez "192.168.166.166" ;
   o Faites OK.

Double-cliquez ensuite sur la ligne "IP_Dst" et indiquez dans "Value :" l'adresse IP de la machine A, où tourne netcat.

6. Modifions les données sur les ports (port par lequel est envoyé le paquet, port de réception sur la machine réceptrice). Pour le port de l'expéditeur (TCP_sport), laissez la valeur à 1024. Pour "TCP_dport" en revanche, indiquez 21 dans "Value :" ;
7. Modifions à présent le numéro de séquence du paquet que nous forgeons ("TCP_seq") et donnons-lui comme valeur "500000" :
8.

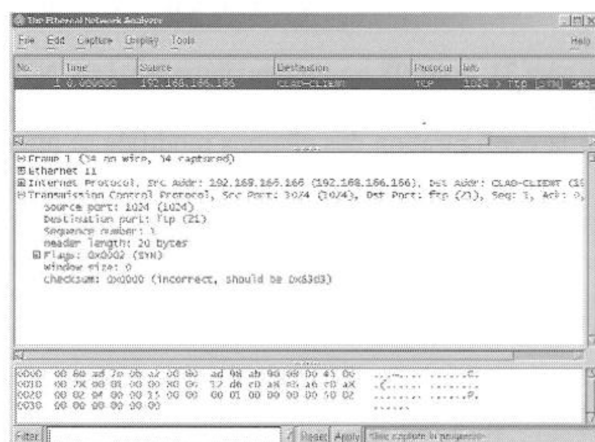


Dans cet exemple la machine A a pour IP 192.168.0.1

9. Tout en bas pour "TCP_cksum", indiquez "Hex" comme "Format" et dans "Value :" mettez 0 ;
10. Vous aurez peut-être remarqué les flags, ou seul le flag SYN est activé ("TCP_syn" est à 1).

Votre premier paquet est forgé. Avant de l'envoyer, même si vous le faisiez maintenant, vous n'obtiendrez rien, corrigeons-en le checksum.

1. Prenez Ethereal ;
2. Lancez une session de capture en mode capture des trames TCP ("filter" ayant pour données "tcp");
3. Injectez le paquet avec Winject ("Inject Packet") ;
4. Regardez sur Ethereal votre paquet qui est passé, ne stoppez pas la capture ;



5. Sélectionnez le paquet en question, développez l'arborescence des données liées à l'en-tête TCP du paquet. En bas du paquet s'affiche la valeur pour checksum que nous avions indiquée à Winject (0), et Ethereal nous donne la valeur qu'il aurait dû prendre "0x83d3") ;
6. Ainsi munis de cette information, dans Winject nous modifions le paquet forgé et dans "TCP_cksum" nous mettons comme "Value :" "83d3" (toujours en format hexadécimal, "HEX") ;

7. On réinjecte le paquet, et Ethereal, qui sniffe toujours, doit vous afficher quelque chose de similaire :

```
2 82.762263   192.168.166.166         CLIENT            TCP   1024 > ftp [SYN] Seq=500000 Ack=0
3 82.762746       CLIENT          192.168.166.166       TCP   ftp > 1024 [SYN, ACK] Seq=3269500
4 85.694229       CLIENT          192.168.166.166       TCP   ftp > 1024 [SYN, ACK] Seq=3269500
5 91.693845       CLIENT          192.168.166.166       TCP   ftp > 1024 [SYN, ACK] Seq=3269500
6 103.693140      CLIENT          192.168.166.166       TCP   ftp > 1024 [SYN, ACK] Seq=3269500
```

Vous devriez très nettement voir apparaître le paquet spoofé corrigé au niveau du checksum être parti vers A, ici "Client", et voir ensuite A tenter de répondre désespérement à 192.168.166.166 ;

Le dénouement approche... La dernière étape va consister à renvoyer un paquet ACK dans le laps de temps imparti avant que A ne considère que la tentative de connexion est un échec, auquel cas le paquet sera refusé. Donc préparons ce paquet avant de lancer l'assaut final.

1. Dans Winject cliquez sur "New" dans la case "Packet" ;
2. Allez sur l'onglet "noname" et cliquez sur "Open" toujours dans la case "Packet" ;
3. Ouvrez Syn.pkt ;
4. Réappliquez quasiment la même configuration du paquet que précédemment, à savoir que les champs modifiés deviennent :
   - o IP_Src : 192.168.166.166
   - o IP_Dst : l'adresse IP de votre machine A, où tourne Netcat sur le port 21
   - o TCP_dport : 21
   - o TCP_seq : 500000
   - o TCP_syn : 0
   - o TCP_ack (pour le champ où se situe "b: 1") : 1
   - o TCP_cksum : 0

Notez que si, à ce stade du cours, vous ne savez pas à quoi servent ces champs, vous pouvez tout reprendre depuis le début.

Vous aurez compris, en ayant suivi le principe tout au long de l'exercice, et en ayant lu la configuration juste au-dessus, que l'objectif va être d'envoyer ce second paquet TCP avec le flag ACK activé dans le temps mort où la machine A essaie de contacter 192.168.166.166 en corrigeant le checksum après avoir modifié la valeur pour le champ ACK renvoyé par A (au niveau de Winject, ce champ sur 32 bits s'appelle "TCP_ack pour le champ où se situe "b: 32"). Donc, avant de pratiquer, voyons comment l'on va pratiquer, et cela va demander une petite répétition des étapes de votre part. A ce stade du processus, vous devriez avoir comme configuration globale :

- Une machine A, sur laquelle tourne netcat sur le port 21, sous 98 ou un deuxième système NT ;
- Une machine B, sur laquelle tourne Ethereal, prêt à la capture mais non actif, et Winject configuré avec 2 paquets (visualisables par 2 onglets) dont la configuration des champs modifiés se présente comme suit :

  - o PACKET 1 :
    - IP_Src : 192.168.166.166
    - IP_Dst : l'adresse IP de votre machine A, où tourne Netcat sur le port 21
    - TCP_dport : 21
    - TCP_seq : 500000
    - TCP_cksum : le checksum calculé par Ethereal.
  - o PACKET 2 :
    - IP_Src : 192.168.166.166
    - IP_Dst : l'adresse IP de votre machine A, où tourne Netcat sur le port 21
    - TCP_dport : 21
    - TCP_seq : 500001
    - TCP_syn : 0
    - TCP_ack/b:1 : 1
    - TCP_cksum : à calculer, donc à 0

© 2001-2002-2003 **DMP**

Ce qui fait que votre premier paquet sur le premier onglet est à 100% opérationnel, le deuxième n'attend plus qu'un numéro d'ACK où le champ est de 32 bits, et un checksum valide

- Si vous ne remplissez pas toutes les conditions nécessaires, faites le nécessaire.

Ce que vous allez faire va être donc - ne le faites pas tout de suite ! - d'envoyer le premier paquet marqué SYN à la machine A, de sniffer la réponse de retour et d'en copier la valeur ACK sur le deuxième paquet, de l'injecter, de le sniffer pour avoir le checksum valide, de corriger le checksum et de l'injecter.
Vous prendrez conscience du résultat si vous activez netcat par la commande "nc -L -v -v -p 21".

Si vous êtes au point pour cet exercice, alors allez-y, bonne chance !

4 - Remarque sur le blind spoofing

La configuration précédente était, on peut le dire, idéale, dans la mesure où le spoofing qui a été réalisé avait lieu en réseau local. Par ailleurs, pour ne rien gâcher, les ordinateurs étaient reliés sur le lan par un Hub. De ce fait, sniffer était simple et l'on pouvait aisément sniffer les paquets transitant sur le réseau et ainsi récupérer le fameux numéro de séquence du paquet Syn/ACK. Dans d'autres circonstances, les choses ne seront pas tout aussi simples. Comment un pirate pourrait-il, dès lors, obtenir ce fameux numéro ?
Si le pirate est sur le même lan que le serveur qui vous intéresse, la solution peut consister en l'exploitation des protocoles de routages tels que RIP ou OSPF. Nous ne détaillerons pas, mais sachez qu'il vaut mieux désactiver ces protocoles si vous n'en avez pas l'utilité. Si vous êtes en réseau switché, le pirate devra utiliser l'arp spoofing, que vous pouvez détecter à l'aide du programme arpwatch.

Dans quel cas est-il utile pour un pirate d'utiliser l'IP Spoofing ? Disons qu'il est surtout intéressant pour passer des limitations basées sur les adresses IPs, à savoir essentiellement les firewalls. Partant de là, comment ferait un pirate pour passer un firewall qui protègerait un serveur sur Internet en n'autorisant que certaines IPs à se connecter ? Il est évidemment hors de question de supposer que les paquets, donc l'ISN qu'il nous faut, lui parviennent, puisqu'il n'est pas sur le réseau local. Il va donc devoir deviner ce numéro de séquence...
Ce qu'il faut bien se dire, c'est que les numéros de séquences ou ISN sont générés 'aléatoirement'. L'aléa n'ayant pas de sens dans un monde fait de déterminisme logique tel que celui de l'informatique, il faut bien se dire que les générateurs de nombres dits aléatoires ne permettent pas en réalité un aléa total. Tout dépendra en effet de la façon dont la pile TCP/IP aura été codée. Vous l'aurez compris, deux systèmes d'exploitations différents n'auront pas les mêmes capacités en matière de génération de nombres aléatoires et, par conséquent, pas les mêmes facultés de résistivité au spoofing. L'utilitaire nmap, qui est une référence dans le monde unix et maintenant windows, permet de donner un ordre de grandeur de la difficulté avec laquelle un système sera spoofable. Vous pourrez donc vous faire une idée de la vulnérabilité de vos équipements réseau.

Pour cela, tapons : nmap -v -O <ip>
Concrètement, nous lançons un fingerprinting couplé à un scan sur l'IP spécifiée puis en fonction des ISN déterminées, nmap donnera une estimation de la difficulté. Comme nous le voyons sur les captures ci dessous, le résultat peut varier énormément d'un système à un autre.

Capture relative à un scan de linux 2.4.x :

```
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=4784921 (Good luck!)

Sequence numbers: 5A83C3D6 5A83C3D6 5B3ACC17 5B3ACC17 5B3CD92D 5B3CD92D
```

Capture relative à un scan de win98 :

```
TCP Sequence Prediction: Class=trivial time dependency
                         Difficulty=3 (Trivial joke)

Sequence numbers: 1678C 167A0 167B4 167C7 167E1 167F9
```

On constate ici que les ISN sont ridiculement prévisibles dans le temps sous win98 (linéarité des ISN) alors que sous linux, la chose bien plus délicate. A noter que sous BSD, les choses sont encore bien plus délicates puisque tout est fait pour engendrer un véritable aléa car les générateurs prennent des bits vrai-

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

ment imprévisibles dans leur environnement : activité du clavier, du processeur, du disque dur, etc. C'est également le cas sur les derniers noyaux linux, très résistants à l'IP spoofing.

La seule solution dans ce contexte est la création de statistiques destinées à essayer de prévoir ces fameuse ISN. Le pirate essaiera, lors d'une attaque, d'envoyer les paquets contenants les ISN ayant statistiquement le plus de chances de tomber. Plus votre bande passante est importante, plus votre réseau est exposé !

VI - Idle Host Scan avec Hping

Dans la partie précédente, nous avons pris connaissance de ce qu'était l'IP spoofing. Nous nous proposons maintenant de voir comment un pirate peut réussir à l'utiliser pour scanner un serveur sans laisser son adresse IP dans les logs. Le but est de prouver que les IPs contenues dans vos fichiers de logs peuvent ête falsifiées, et de vous inciter quand c'est possible à mettre en place des règles de filtrage pour qu'aucun "idle host" n'ait accès au serveur, ce qui empêchera la mise en oeuvre de cette technique de scan. Etudions la chose de plus près...

Idle host scan :

L'idle host scan est une technique trouvée par le créateur de Hping qui permet de scanner les ports d'un serveur, sans laisser son IP dans ses logs, et en utilisant l'IP spoofing. Mais pour cela, il nous faut une machine peu active qui va nous servir pour déterminer les ports ouverts du serveur... Vous allez comprendre ;)

Simulation pratique d'une telle attaque :

Nous avons utilisé trois machines sur nos serveurs de la Hackademy :

- Une machine attaquante sous linux ou équivalent Unix (car Hping ne tourne que sous ces systèmes) --> que nous appellerons A
- Un serveur cible --> que nous appellerons S
- Une machine peu active (surtout pas un serveur ! :) --> à savoir C

La machine A doit être sous linux ou équivalent et avoir installé Hping que l'on peut télécharger sur http://www.hping.org. Dans un premier temps, ce qui va nous intéresser, sera de contrôler l'incrémentation du numéro d'identification (id) du protocol IP des paquets émis par C. Pour cela, deux conditions doivent être réunies : nous devons forcer la machine C à nous envoyer des paquets tcp/ip (quels qu'ils soient) de façon continue, et C ne doit avoir aucune connection active pendant la durée des opérations (excepté avec A, cela va sans dire). Nous comprenons tout de suite l'intérêt de choisir une machine client pour C. Celle-ci n'étant pas serveur, il y a plus de chances que nous soyons les seuls à communiquer avec elle.

Pour dialoguer avec C et lui forcer à nous révéler son id, deux solutions sont envisageables :

1) Nous essayons d'initialiser une connection sur un port actif de la machine C en envoyant des paquets avec le flag Syn actif (très important, sinon la machine ne répondra pas). La machine nous répondra alors par des paquets avec les flags Syn et Ack activés.

2) Nous envoyons des paquets sur un port fermé (pas besoin d'activé le moindre flag). La machine croira à une erreur et nous enverra des paquets réponses avec les flags Ack et Reset actifs.

Quelle méthode est la plus valable ? Si les deux fonctionnent très bien, la première présente deux désavantages de taille : si l'on veut rester logique avec nos propos antérieurs, choisir une machine C qui n'aurait ne serait-ce qu'un port ouvert est ridicule (c'est un serveur dans ces cas-là et les risques de connections avec un autre que A sont importantes). Par ailleurs, envoyer des flots de demandes de connection sur C sur un port précis pourrait être mal interprété par un administrateur consciencieux qui regarderait ses logs (il pourrait croire à un syn flooding ou à un syn scanning).

Nous allons donc forger des paquets selon le second schéma, et pour cela, nous utiliserons hping. Hping, tout comme nemesis, est un forgeur de paquets tcp/ip. Pourquoi choisir hping ? Et bien, hping possède une fonctionalité intéressante qui est celle de loguer sur la console les réponses aux paquets émis via hping. Si nous avions utilisé nemesis par exemple, il eût fallu utiliser un sniffer pour regarder les paquets émis.

© 2001-2002-2003 DMP

Il nous suffira de taper la commande suivante :
hping <ip de C> -r
(par ex : hping 192.168.1.5 -r)

Ici, -r correspond à l'option montrant l'incrément de l'id au fur et à mesure des émissions (on ne choisira cette option que pour des raisons de clarté).

```
xterm
kheops:/home/toki# hping 192.168.1.5 -r
HPING 192.168.1.5 (eth0 192.168.1.5): NO FLAGS are set, 40 headers + 0 data byt
es
len=46 ip=192.168.1.5 flags=RA seq=0 ttl=128 id=47878 win=0 rtt=11.0 ms
len=46 ip=192.168.1.5 flags=RA seq=1 ttl=128 id=+256 win=0 rtt=2.0 ms
len=46 ip=192.168.1.5 flags=RA seq=2 ttl=128 id=+256 win=0 rtt=0.5 ms
len=46 ip=192.168.1.5 flags=RA seq=3 ttl=128 id=+256 win=0 rtt=0.4 ms
len=46 ip=192.168.1.5 flags=RA seq=4 ttl=128 id=+256 win=0 rtt=0.5 ms

--- 192.168.1.5 hping statistic ---
5 packets tramitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.4/2.9/11.0 ms
kheops:/home/toki#
```

Voici une capture réalisée avec Ethereal des paquets envoyés et reçus successivement entre les machines A et C.
1er paquet : Dans la capture, la machine A est celle qui a l'IP 192.168.1.2 et la machine C est celle qui a l'IP 192.168.1.5.

```
⊟ Internet Protocol, Src Addr: kheops (192.168.1.2), Dst Addr: 192.168.1.5 (192.168.1.5)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 40
    Identification: 0x1e34
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xd944 (correct)
    Source: kheops (192.168.1.2)
    Destination: 192.168.1.5 (192.168.1.5)
```

C'est le protocole IP qui va surtout nous intéresser :). Le premier paquet est donc envoyé par la machine A vers la machine C. C'est un paquet TCP envoyé sur le port 0 de la machine C, avec tous les flags à 0. Avec un numéro de séquence et numéro d'acquittement. Notez bien qu'ici l'Identification IP est égale à 0x1e34 (1e34 en hexadécimal).

2ème paquet :

```
⊟ Internet Protocol, Src Addr: 192.168.1.5 (192.168.1.5), Dst Addr: kheops (192.168.1.2)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 40
    Identification: 0xbb06
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0xfc71 (correct)
    Source: 192.168.1.5 (192.168.1.5)
    Destination: kheops (192.168.1.2)
```

La machine C répond avec un paquet TCP contenant RST/ACK à 1. (normal le port 0 n'est pas ouvert). Notez que son numero d'Identification IP est 0xbb06

3ème paquet :

```
⊟ Internet Protocol, Src Addr: kheops (192.168.1.2), Dst Addr: 192.168.1.5 (192.168.1.5)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 40
    Identification: 0x06a0
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xf0d8 (correct)
    Source: kheops (192.168.1.2)
    Destination: 192.168.1.5 (192.168.1.5)
```

La machine A renvoie un paquet TCP sur le port 0 avec tous les flags à 0.

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

Notez que notre numéro d'Identification IP s'est incrémenté de plusieurs bits passant de 0x1e34 à 0x06a0

4ème paquet :
La machine répond avec un paquet TCP identique au 2ème paquet, et regardez le numéro d'identification IP ! Il s'est incrémenté de +256 !

```
⊟ Internet Protocol, Src Addr: 192.168.1.5 (192.168.1.5), Dst Addr: kheops (192.168.1.2)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 40
    Identification: 0xbc06
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0xfb71 (correct)
    Source: 192.168.1.5 (192.168.1.5)
    Destination: kheops (192.168.1.2)
```

En effet, si l'on compare l'identification IP avec celle du 2ème paquet, on peut remarquer qu'il passe de 0xbb06 à 0xbc06 (faites le calcul si vous ne me croyez pas :p).

3ème, 4ème, 5ème, 10ème paquets : Regardez ! Le numéro d'identification de la machine A s'incrémente toujours de +256 ! Attention, ce n'est pas tant la valeur de l'incrément qui nous intéresse ici mais bel et bien le fait qu'il soit constant. C'est grâce à ce numéro d'identification que l'on va pouvoir déterminer si un port est ouvert sur le serveur S.

Maintenant, la machine A doit envoyer un paquet spoofé vers le serveur S, pour lui faire croire que la machine C désire se connecter. Pour mettre en place le scan, on garde la première fenêtre ouverte et on en ouvre une dans laquelle on tapera :
hping -a <IPspoofée> -S -p <port> <ipdestination>
Ex : hping -a 192.168.1.5 -S -p 21 192.168.1.1 où 21 est un port dont nous allons essayer de déterminer s'il est ouvert ou non.

On lancera les émissions de paquets en simultané pour une meilleure analyse des résultats.

Premier terminal : émission des paquets spoofés au serveur S.

```
xterm
kheops:/home/toki# hping -a 192.168.1.5 -S -p 21 193.251.61.27
HPING 193.251.61.27 (eth0 193.251.61.27): S set, 40 headers + 0 data bytes
len=46 ip=193.251.61.27 flags=SA DF seq=0 ttl=126 id=51804 win=16616 rtt=90.4 m
len=46 ip=193.251.61.27 flags=SA DF seq=1 ttl=126 id=51805 win=16616 rtt=89.8 m
len=46 ip=193.251.61.27 flags=SA DF seq=2 ttl=126 id=51806 win=16616 rtt=88.8 m
len=46 ip=193.251.61.27 flags=SA DF seq=3 ttl=126 id=51808 win=16616 rtt=90.7 m
len=46 ip=193.251.61.27 flags=SA DF seq=4 ttl=126 id=51809 win=16616 rtt=89.6 m
len=46 ip=193.251.61.27 flags=SA DF seq=5 ttl=126 id=51810 win=16616 rtt=88.7 m
len=46 ip=193.251.61.27 flags=SA DF seq=6 ttl=126 id=51811 win=16616 rtt=90.9 m

--- 193.251.61.27 hping statistic ---
8 packets tramitted, 7 packets received, 13% packet loss
round-trip min/avg/max = 88.7/89.8/90.9 ms
kheops:/home/toki# 
```

Ici, C demande des connections à S. L'ID s'incrémente de +1 à chaque envoi. Notez le fait que nous percevons les paquets de C sur notre terminal. Ceci ne s'est produit que parce que nous travaillons dans un lan doté d'un hub. Par conséquent, les paquets de C nous reviennent. Si C avait eu une IP publique (internet) nous n'aurions aucun moyen de récupérer les paquets. Notons toutefois que ce n'est absolument pas indispensable. Une information importante pour la compréhension est contenue dans ces paquets. Les paquets de retour ont les flags Syn et Ack à 1. Par conséquent, le port 21 est ouvert.

Deuxième terminal : émission des paquets TCP/IP au client C.

```
xterm
es
len=46 ip=192.168.1.5 flags=RA seq=0 ttl=128 id=56838 win=0 rtt=0.6 ms
len=46 ip=192.168.1.5 flags=RA seq=1 ttl=128 id=+256 win=0 rtt=0.4 ms
len=46 ip=192.168.1.5 flags=RA seq=2 ttl=128 id=+512 win=0 rtt=0.5 ms
len=46 ip=192.168.1.5 flags=RA seq=3 ttl=128 id=+512 win=0 rtt=0.4 ms
len=46 ip=192.168.1.5 flags=RA seq=4 ttl=128 id=+512 win=0 rtt=0.3 ms
len=46 ip=192.168.1.5 flags=RA seq=5 ttl=128 id=+512 win=0 rtt=0.4 ms
len=46 ip=192.168.1.5 flags=RA seq=6 ttl=128 id=+512 win=0 rtt=0.3 ms
len=46 ip=192.168.1.5 flags=RA seq=7 ttl=128 id=+512 win=0 rtt=0.4 ms
len=46 ip=192.168.1.5 flags=RA seq=8 ttl=128 id=+512 win=0 rtt=0.4 ms
len=46 ip=192.168.1.5 flags=RA seq=9 ttl=128 id=+512 win=0 rtt=0.4 ms
len=46 ip=192.168.1.5 flags=RA seq=10 ttl=128 id=+256 win=0 rtt=0.3 ms
len=46 ip=192.168.1.5 flags=RA seq=11 ttl=128 id=+256 win=0 rtt=0.4 ms

--- 192.168.1.5 hping statistic ---
12 packets tramitted, 12 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.4/0.6 ms
kheops:/home/toki# 
```

Regardons en parallèle ce qui se trouve sur notre deuxième terminal, le seul qui présente un intérêt ici.
Nous observons que les id des paquets sont tous incrementés de +256 mais à partir du 3ème, l'incrément est doublé (+512). L'ID est doublé lors de l'émission de 8 paquets (seq=2 à seq=9) exactement. Or 8 paquets exactement avaient été émis (voir bas de la capture précédente). Que peut-on en conclure ? Puisque l'ID a doublé, cela ne peut signifier qu'une seule chose : 8 paquets ont été émis par C mais qui n'étaient pas à destination de A, d'où le saut d'incrément de +256 à +512. Quels étaient ces paquets ? Si vous vous référez au chapitre

© 2001-2002-2003 **DMP**

précédent, il s'agissait nécessairement de paquets de C vers S avec le flag RST actif. Pourquoi donc ? Eh bien C n'a jamais lui-même demandé de connection. Il informe donc S à chaque paquet avec un Syn / Ack de S qu'il reçoit qu'il s'agit d'une erreur.

On sait que, sur le serveur qui nous intéresse, le port 22 est fermé. On va donc répéter la même expérience en troquant le port 21 pour le port 22. Observons les résultats.

Premier terminal : émission des paquets spoofés au serveur S

```
xterm
kheops:/home/toki# hping 192.168.1.5 -r
HPING 192.168.1.5 (eth0 192.168.1.5): NO FLAGS are set, 40 headers + 0 data byt
es
len=46 ip=192.168.1.5 flags=RA seq=0 ttl=128 id=61958 win=0 rtt=24.1 ms
len=46 ip=192.168.1.5 flags=RA seq=1 ttl=128 id=+256 win=0 rtt=0.7 ms
len=46 ip=192.168.1.5 flags=RA seq=2 ttl=128 id=+256 win=0 rtt=0.5 ms
len=46 ip=192.168.1.5 flags=RA seq=3 ttl=128 id=+256 win=0 rtt=0.6 ms
len=46 ip=192.168.1.5 flags=RA seq=4 ttl=128 id=+256 win=0 rtt=0.5 ms
len=46 ip=192.168.1.5 flags=RA seq=5 ttl=128 id=+256 win=0 rtt=0.5 ms
len=46 ip=192.168.1.5 flags=RA seq=6 ttl=128 id=+256 win=0 rtt=0.8 ms
len=46 ip=192.168.1.5 flags=RA seq=7 ttl=128 id=+256 win=0 rtt=0.5 ms
len=46 ip=192.168.1.5 flags=RA seq=8 ttl=128 id=+256 win=0 rtt=0.5 ms
len=46 ip=192.168.1.5 flags=RA seq=9 ttl=128 id=+256 win=0 rtt=1.3 ms

--- 192.168.1.5 hping statistic ---
10 packets tramitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 0.5/3.0/24.1 ms
kheops:/home/toki#
```

On note que les paquets de retour n'ont plus les flags Syn et Ack à 1 mais Ack et Reset. Logique, les ports sont fermés.

Deuxième terminal : émission des paquets TCP/IP au client C.

```
Xterm
kheops:/home/toki# hping -a 192.168.1.5 -S -p 22 193.251.61.27
HPING 193.251.61.27 (eth0 193.251.61.27): S set, 40 headers + 0 data bytes
len=46 ip=193.251.61.27 flags=RA seq=0 ttl=126 id=51837 win=0 rtt=91.0 ms
len=46 ip=193.251.61.27 flags=RA seq=1 ttl=126 id=51838 win=0 rtt=89.0 ms
len=46 ip=193.251.61.27 flags=RA seq=2 ttl=126 id=51839 win=0 rtt=90.8 ms
len=46 ip=193.251.61.27 flags=RA seq=3 ttl=126 id=51840 win=0 rtt=89.8 ms
len=46 ip=193.251.61.27 flags=RA seq=4 ttl=126 id=51841 win=0 rtt=88.8 ms
len=46 ip=193.251.61.27 flags=RA seq=5 ttl=126 id=51842 win=0 rtt=91.0 ms
len=46 ip=193.251.61.27 flags=RA seq=6 ttl=126 id=51843 win=0 rtt=89.6 ms
len=46 ip=193.251.61.27 flags=RA seq=7 ttl=126 id=51844 win=0 rtt=88.6 ms
len=46 ip=193.251.61.27 flags=RA seq=8 ttl=126 id=51845 win=0 rtt=89.4 ms
len=46 ip=193.251.61.27 flags=RA seq=9 ttl=126 id=51846 win=0 rtt=92.0 ms

--- 193.251.61.27 hping statistic ---
10 packets tramitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 88.6/90.0/92.0 ms
kheops:/home/toki#
```

Ici, on constate que l'ID n'a pas changé. Pourquoi me demanderez-vous ? Tout simplement parce qu'une machine ne repond jamais à un Reset. D'ailleurs, soyons logiques, si une machine devait y répondre, ce serait nécessairement par un Reset aussi. Donc, la machine renverrait une Reset une deuxième fois et ainsi de suite.
Lorsque le port est fermé, il n'y pas plus "fuite" de paquets, donc plus de modifications de l'ID au cours du temps. Vous comprenez sans doute maintenant à quel point il est vital que la machine C ne corresponde qu'avec A.

Conclusion :

Dans ce chapitre, nous avons vu le principe de fonctionnement de ce type de scan.

A noter que les unixiens utilisent le célèbre scanner de port de fyodor (nmap) (à télécharger sur http://www.insecure.org) qui prend en compte ce type de scan qui peut s'avérer redoutable. Remarquez que même si vous n'êtes pas dans les logs de S, vous êtes dans ceux de C. Un pirate peut donc être retrouvé...

Remarque :

- Rappelons que le scan de port consitue un délit. Vous n'êtes autorisés à scanner que vos propres machines en réseau local.
- Le(s) IP publique(s) utilisées pour réaliser ce chapitre sont celles des professeurs de The Hackademy. En tant que tel et avec leur accord, nous n'avons enfreint aucune règle. Vous n'êtes en aucun cas autorisés à scanner ces IP.
- Dans cette démonstration, nous avons utilisé l'ip 192.168.1.5 pour C et 192.168.1.2 pour A. Il est bien évident que le choix de ces IP (IP privées) n'est intelligent que dans le cadre de démonstrations. Un pirate serait démasqué selon toute logique s'il commettait l'erreur de choisir la machine spoofée dans son lan (A et C ont la même IP publique dans ces cas).

© 2001-2002-2003 **DMP**

VII - Les network mappers

Bien qu'il soit utile de consacrer un chapitre à ce type de logiciels, car c'est bien de logiciel dont nous allons parler, nous ne diviserons les éléments qu'en trois parties : une partie sur le principe de ce type de logiciels, et leur utilité, et une autre sur la présentation, et quelques principes d'utilisation, de WhatsUP Gold Net Monitor d'IPSwitch, un puissant network mapper.

1 - Principe du network mapper et utilité :

Le principe de base du network mapper, c'est la cartographie. Les éléments à cartographier varient selon les networks mappers, mais en informatique, ce sont généralement des machines telles que des serveurs, des stations de travail, des routeurs, des passerelles, etc.

Le network mapper va permettre de regrouper tout un ensemble d'informations sur les ensembles de machines qui forment des réseaux. D'un point de vue commercial, ce logiciel est véritablement l'outil typique de surveillance et d'analyse dont peut avoir besoin un administrateur réseau, que ce soit pour surveiller depuis un seul poste que toutes les machines se portent bien sur le réseau, ou encore pour être alerté des services qui tombent en panne...

Mais un pirate informatique pourrait faire une "mauvaise" utilisation de ce type de logiciels. Nous allons voir comment, tourné vers des cibles sur Internet et avec ce type de logiciels, un pirate pourrait organiser une attaque structurée et élaborée envers le réseau d'ordinateurs d'une entreprise accessibles depuis Internet. La force de l'attaque dépend ensuite de la facilité pour le pirate à s'organiser, à monter des stratégies efficaces, à utiliser le logiciel. Le résultat dépend ensuite de ses compétences techniques globales.

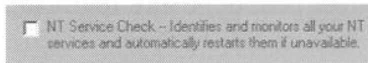
Relativisons toutefois en précisant que le plus grand nombre d'attaques envers des systèmes se font sur des cibles localisées, souvent individuelles. Cependant, un pirate peut souhaiter s'attaquer à une entreprise et avoir recours à un Network Mapper pour fonder ses actions. Bien utilisé, ce logiciel peut garantir trois fois plus d'efficacité sur les actions d'un pirate. Ce qui freine ce genre d'activités, c'est aussi le temps que peut prendre la récolte des informations avant analyse et surveillance, et leur organisation au sein des données que peut enregistrer un network mapper.

2 - Installation, configuration d'un Network mapper :

Rendez-vous sur le site http://www.ipswitch.com/. Allez dans "Download" où vous pourrez télécharger une version d'évaluation de WhatsUp Gold, d'à peu près 10 Mbits.

Après exécution de l'installation, procédez aux étapes courantes nécessaires, mais prêtez attention à ces étapes-ci de l'installation :

1. Laissez décochée la case "NT Service Check" ;



2. Idem pour la case "Syslog" qui vient juste après ;
3. Poursuivez le processus d'installation jusqu'à avoir la fenêtre de lancement de la version d'évaluation du logiciel.

Au lancement du logiciel, allez dans l'onglet "Configure". Il n'y a pas de vraie configuration du logiciel à faire. Notez toutefois que vous pouvez y configurer un server HTTP sur votre machine, dans "Web Server", ou même certaines caractéristiques du logiciel, dans "Program Options" dont on retiendra :

- Possibilité de gérer un fichier de log ("logging") ;
- Possibilité de gérer les icônes sur carte dont nous allons aborder le principe ("Device States"), ou de la carte elle-même ("Map Settings") ;

Rien de très compliqué à gérer au niveau de la configuration, les spécifications d'utilisation s'effectuant plus tard, à travers les outils proposés par le logiciel pour gérer au mieux vos données. En effet, un Network Mapper ne se contente pas uniquement de créer une carte pour placer les machines dont vous avez les adresses IP. Il intègre souvent une suite logicielle d'utilitaires dédiés à la résolution de services réseaux (scanning de ports), à la communication avec ceux-ci (divers utilitaires supportant des protocoles tels

© 2001-2002-2003 **DMP**

SNMP ou encore permettant de transférer des informations avec des DNS), et surtout, point crucial, à la surveillance de leur activité. Non, vous ne saurez pas qui se connecte ou ne serez pas en mesure de capturer quelconque donnée, mais vous pourrez voir, 24h/24 l'ensemble de l'activité réseau (machines qui se déconnectent, ne répondent plus, ont des services qui tombent en panne ou déménagent, etc). Capacité native très utile puisqu'elle va permettre de ne rien laisser passer sur les systèmes sous écoute...

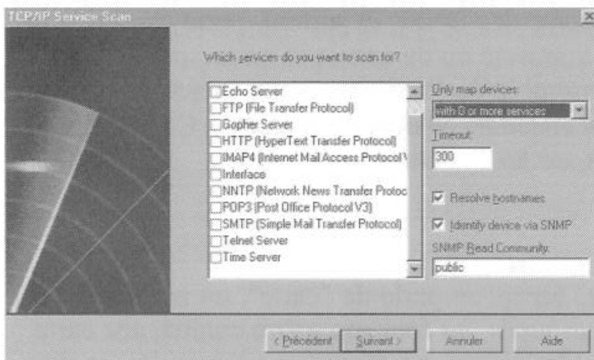
3 - Utilisation d'un network mapper :

Vous aurez compris, par les explications précédentes, que lorsqu'on parle de "carte", on parle de l'ensemble des données que gèrent les networks mappers, qui, regroupés, les font apparaître comme une carte. N'oubliez pas que ces logiciels sont, à la base, destinés aux administrateurs réseaux en entreprise, et que, par conséquent, il leur sert plus à monter des schémas liés à des outils de gestion réseau de leurs systèmes, qu'à pirater.

Il existe un outil de création de cartes sous WhatsUp qui passe par différentes étapes de configuration et d'utilisation et qui peut être tout aussi utile que de partir d'une carte créée à vide. Cet outil va vous permettre de monter votre carte avec une configuration propre et avec l'aide d'utilitaires réseaux. Voyons cela tout de suite :

1. Allez dans l'onglet "File" ;
2. Cliquez sur "New Map Wizard" ;
3. Choisissez "Discover and map network devices" ;
4. Sur la fenêtre suivante s'affiche différentes options qui vont servir à découvrir votre réseau local, pour le cartographier. Comme les pirates se tournent vers des cibles Internet, il leur faut adapter leur configuration pour des analyses vers des cibles Internet. Ainsi, ils choisiront de ne laisser cochée que la case : "Discover your Network using ICMP", ce qui veut dire "Découvrir le réseau avec le protocole ICMP" (ICMP, protocole de résolution d'adresse IP utilisé pour les pings notamment). C'est, ni plus ni moins, ce qu'il faut pour scanner la présence de moult machines sur Internet. Adaptez donc votre configuration comme vous l'entendez après avoir pris note de ce que l'on a vu ci-dessus.
5. Sur la fenêtre suivante, vous devez indiquer une adresse IP où va commencer le scanning (pour résoudre la présence des machines), et l'adresse IP où il va se terminer. Un pirate qui visera des machines d'entreprise sur Internet et veut faire une liste de toutes leurs machines onlines, va indiquer une plage d'adresse IP contenant l'ensemble, ou un gros ensemble, des adresses IP des machines de l'entreprise connectées à Internet en faisant un scann de présence (ping) commençant par l'adresse IP la plus basse pour aller à l'adresse IP la plus haute. C'est un peu le même principe que le scanning de plages de port, sauf qu'ici, il faut bien déterminer une plage d'adresses qui ne soit ni trop courte, ni trop longue. WhatsUp intègre des outils (tirés de WS Ping pro Pack notamment) qui vont permettre à un pirate, ou à un administrateur réseau, de faire ce travail efficacement.
   Sachant que votre provider condamne le scanning, et que, même d'un point de vue juridique, la question est épineuse, nous vous déconseillons de pratiquer cette méthode. Vous pourriez voir votre compte fermé ! Par contre, rien ne vous empêche de scanner votre réseau personnel pour voir quelles informations un pirate pourrait obtenir.
   L'outil de scan pour la création de carte de WhatsUp ne nous paraît pas ici assez développé. Il en intègre un autre qui est déjà bien plus pratique, nous le verrons donc plus tard.
   Sur cette fenêtre, indiquez, puisque nous ne scannerons pas par ce biais, à "Start address" la valeur "127.0.0.1" et à "Ending address" la valeur "127.0.0.1" également. Ainsi, vous ne scannerez que votre machine, en partant de votre machine pour arriver à votre machine. On n'avance pas beaucoup, mais c'est démonstratif. Laissez les autres options en l'état, et passez à la fenêtre suivante.
6. Le logiciel affiche ensuite un utilitaire de résolution des services courants, ce qui va permettre, lorsqu'il scannera les machines sur la plage d'adresses IP, de vérifier également si les services sélectionnables ne sont pas actifs. D'un point de vue technique, c'est une simple résolution d'un port ouvert associé au service courant qui l'utilise (port 80 pour le service HTTP par exemple), fait par un paquet TCP, avec le flag SYN activé, en direction du port (service) en question. Vous pourriez tous les sélectionner pour être le plus pointu possible dans votre récupération d'informations, mais comme vous scannez votre machine, et que, encore une fois, cet utilitaire est limité, nous ne cocherons rien.

© 2001-2002-2003 **DMP**



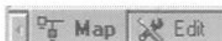
Pour l'option "Only map devices", vous pouvez indiquer de ne cartographier que les machines faisant tourner aucun ou plus de services, au minimum un ou deux services. Un pirate fera une cartographie exhaustive des machines sur le net de l'entreprise, car des services peuvent apparaître ou des informations se dévoiler. Il laissera donc le choix sur "with 0 or more services". Dans Timeout, nous vous conseillons de préciser généralement un plus grand délai ; certains serveurs mettent plus de temps à répondre que d'autres, le délai de transmission peut alors dépasser les 300ms. Laissez activé "resolve hostnames", sachant qu'il va permettre de résoudre les noms d'hôtes des machines, nous avons déjà vu ce dont il s'agissait au début du cours. "Identify device via SNMP" permet de résoudre des services si une machine fait tourner un service SNMP en "public", nous avons aussi vu ça au début des cours. Désactivez l'option, elle risque de ralentir le processus et de ne pas récolter assez d'informations. Passez à l'étape suivante.

1. Le logiciel lance le scan, qui, ici, ne devrait prendre que quelques secondes. L'affichage des résultats ne devrait prendre en compte que votre machine. Cliquez sur "Terminer".

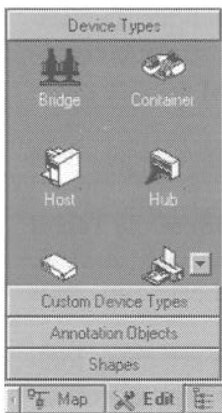
S'est créée alors une carte qui ne contient que votre machine. Tout ça pour juste cette petite icône ? Oui, mais une petite icône qui reflète l'activité sur le système. Si celui-ci se met à ne plus répondre aux paquets ping, alors le logiciel va faire changer la couleur de l'icône afin de vous informer d'une anomalie. En surveillant votre propre système, vous n'aurez jamais l'occasion de voir un tel événement. C'est pourquoi nous allons étudier les rajouts d'éléments sur la carte.

Utiliser les modes d'édition et de surveillance

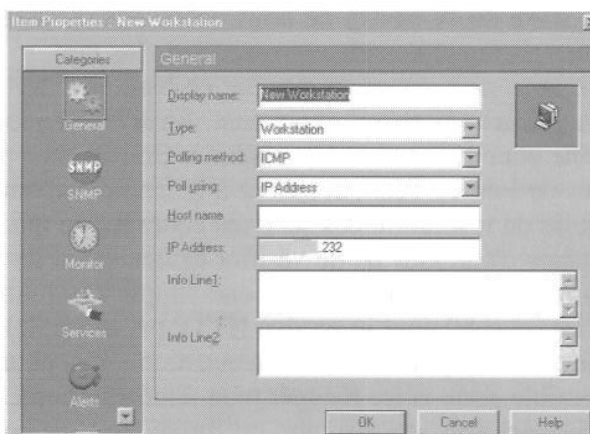
La carte que vous avez sous les yeux est en mode d'analyse de la, ou des, machines qui sont enregistrées. Pour passer en mode d'édition de la fenêtre, utilisez l'onglet situé en bas de la carte :



Les éléments d'édition (icônes) se trouvent à gauche.



Pour rajouter un élément sur la carte, sélectionnez une icône, et faites-la glisser vers l'emplacement voulu sur le support. De là, éditez-en les propriétés (sélectionnez l'élément et faites un clic droit puis cliquez sur "Properties").

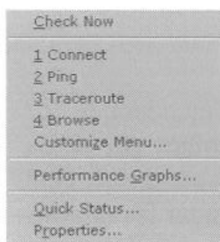


Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**

LES COURS PAR CORRESPONDANCE DE THE HACKADEMY SCHOOL

De là, sachez que vous pouvez modifier le nom de l'icône sur la carte ("Display Name"), indiquer un autre type de machine "Type", modifier la méthode de résolution (laissez sur ICMP)... Par contre, les deux cases suivantes sont plus intéressantes. Ce sont elles qui permettent de spécifier l'adresse IP de la machine à surveiller. Le hostname est facultatif, il fait plus office d'information. Vous pouvez ensuite rajouter des commentaires dans "Info Line". Validez avec OK, quand tout est configuré. Notez éventuellement les outils à votre gauche, qui ne nous serviront pas ici.

Revenu au menu principal retournez sur l'onglet "Map" en-dessous de la carte. L'objet créé est consultable par un simple clic droit (connexion par telnet, ping, traceroute, consultabilité par HTTP), vous pouvez utiliser la fonction "Check Now" pour demander une vérification immédiate de la présence en ligne de la machine.



Et ce n'est pas tout ce qu'on peut dire de la surveillance. Vous en comprendrez l'ampleur en utilisant un outil intégré à WhatsUP : "Net Tools" (repris de Ws Ping pro Pack, adapté à WhatsUp).

Utilisation de Net Tools :

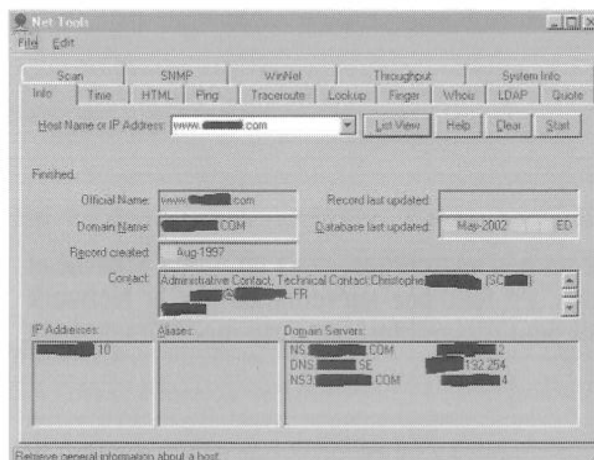
Allez dans l'onglet "Tools" puis dans "Net tools", cliquez. S'ouvre alors une boîte à outil contenant des applications pratiques. Si vous avez déjà manipulé WS Ping Pro Pack (du même concepteur) alors, vous êtes familier avec l'interface. Pour les autres, familiarisez-vous avec elles, nous n'allons voir ici que deux sections du logiciel : "Info" et "Scan".

Nous allons, pour comprendre comment un pirate fait, décrire pas à pas les processus qui sont appliqués pour récolter les informations les plus pertinentes et concises possibles sur des plages d'adresses IP.

L'objectif pour le pirate expérimenté sera d'acquérir ces données, sur lesquelles il fixera ses actions :

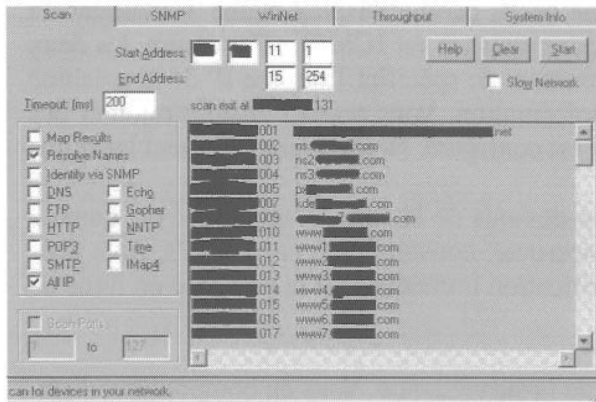
- L'ensemble des machines d'une société X, et uniquement d'une société X, mises en ligne sur Internet. Nous verrons comment le pirate sélectionne ses plages d'adresses IP, et arrive à choisir les bons systèmes.
- La liste exhaustive des services et applications serveurs en place sur toutes les machines. Là aussi, nous verrons quels outils sont à la disposition du pirate pour accomplir la tâche.

Situons déjà un contexte. Un pirate possède l'adresse web d'une société X. Il désire causer du tort à la société pour diverses raisons. Il va tout d'abord acquérir l'adresse IP du serveur web de l'entreprise. Pour cela, il va utiliser Net Tools, aller dans l'onglet "Info". L'utilitaire va se charger de résoudre le nom d'hôte en adresse IP, résoudre les serveurs DNS et faire un whois concernant le site. Que du très banal, sauf qu'à partir de l'adresse IP qu'a le pirate, il va pouvoir scanner les machines alentour.

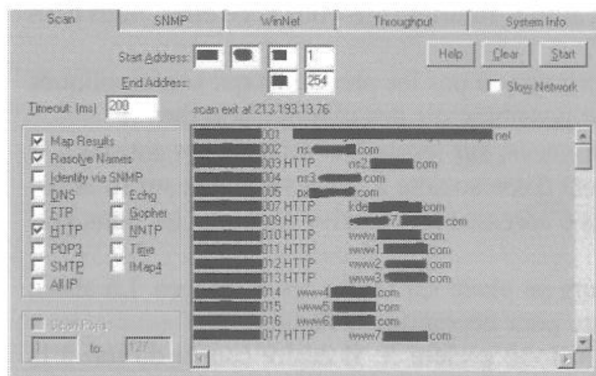


Ici, l'adresse IP xx.xx.xx.10 pourra être réutilisée dans les fonctions de scan. Le pirate se rendra dans l'onglet scan où il réutilisera le début de l'adresse IP du serveur pour définir sa plage d'adresses. Il en définira premièrement une large, puis à partir de là, il va se restreindre à une plage plus courte. Il pourra ensuite faire cartographier ses données sur la carte à construire.

© 2001-2002-2003 **DMP**



Après ce scan préparatif, il s'apercevra que, par exemple, les serveurs du réseau scanné s'étendent de l'IP xx.xx.xx.1 à xx.xx.xx..254. Il va peut-être vouloir choisir uniquement les machines résolues par noms d'hôtes (ce qui élimine beaucoup de stations de travail, et a plutôt tendance à porter sur les serveurs), et, à titre informatif, va vouloir résoudre ceux qui font tourner un service HTTP. Il va donc changer les options de Net Tools, pour que celui-ci ne donne en résultat que des serveurs HTTP, et va faire l'exportation automatique de ces mêmes résultats sur la carte, à l'aide de l'option "Map results"?. Comme sur l'image qui suit :



Les résultats se reportent ensuite immédiatement sur la carte :



Certes, s'il y a trop de machines résolues, tout ceci semble un peu anarchique, mais un pirate acharné et patient scannera l'ensemble des ports de toutes ces machines, en fera une surveillance par le Network Mapper, arrangera les données, les groupera, etc. Voilà à quoi peut ressembler une carte modérément bien construite :

Une publication **The Hackademy School**
© 2001-2002-2003 **DMP**